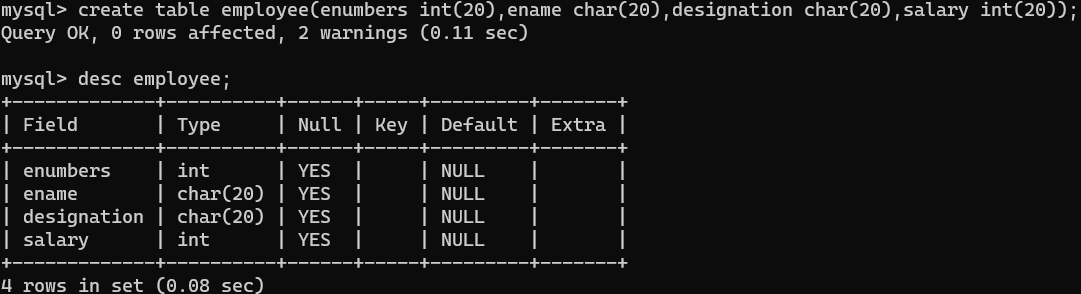
**Author - Mihir Patel**

1. Write a query to create table employee with enumbers , ename , designation and salary.

Syntax :- create table tablename (columnname1 datatype(size),

Columnname2 datatype(size));

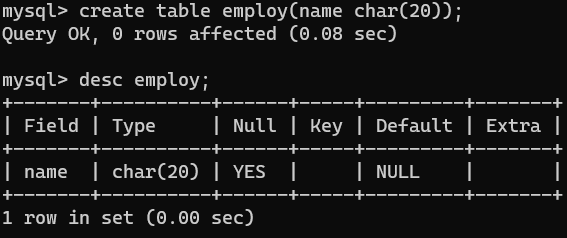
EX :-



1. Write a query to display a column name and datatype of table employ.

Syntax :- desc tablename;

EX :-



1. Write a query to create other table from an existing table with same fields.

Syntax :- create table tablename(new) as select \*from tablename(old);

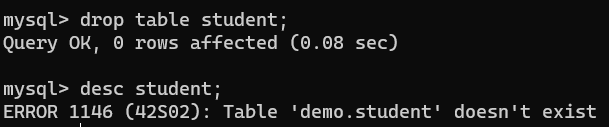
EX :-



1. Write a query to drop a table name student.

Syntax :- drop table tablename;

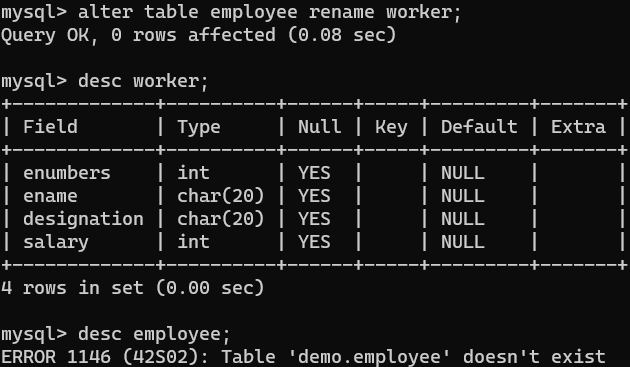
EX :-



1. Write a query to rename table employee to worker.

Syntax :- alter table tablename(old) rename tablename(new);

EX :-



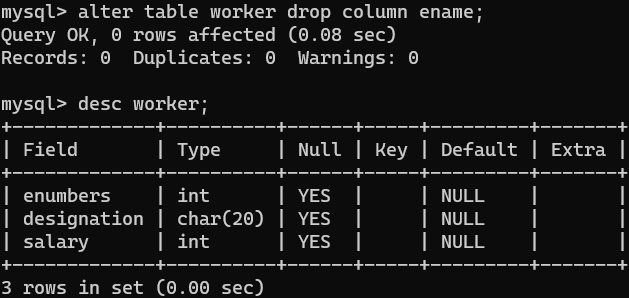
1. Write a query to drop a column from an existing table employ (means worker).

Syntax :-

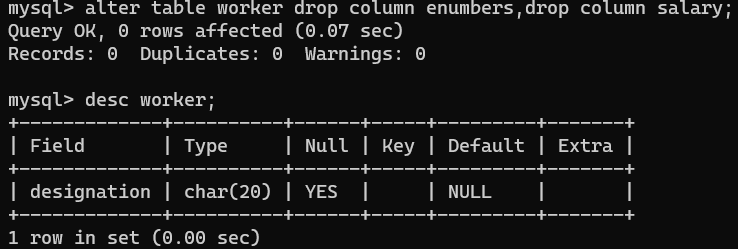
1. Single :- alter table tablename drop column columnname;
2. Multiple :- alter table tablename drop column columnname1,drop column columnname2;

EX :-

Single :-



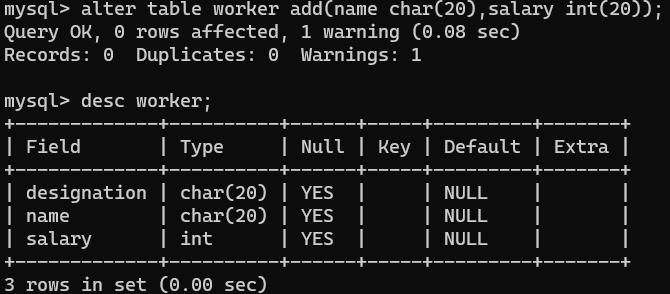
Multiple :-



1. Write a query to add a new column name and salary in employee (means worker) table.

Syntax :- alter table tablename add columnname datatype(size);

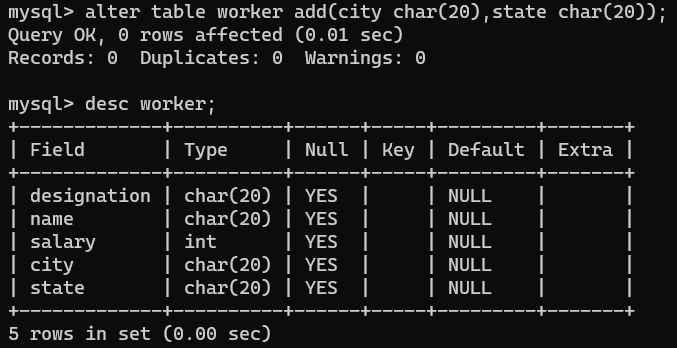
EX :-



1. Write a query to add city and state column to worker table.

Syntax :- alter table tablename add(columnname1 datatype(size),columnname2 datatype(size));

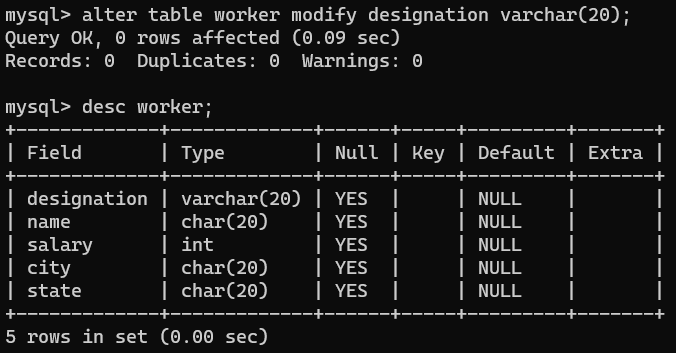
EX :-



1. Write a query to change datatype.

Syntax :- alter table tablename modify columnname datatype(size);

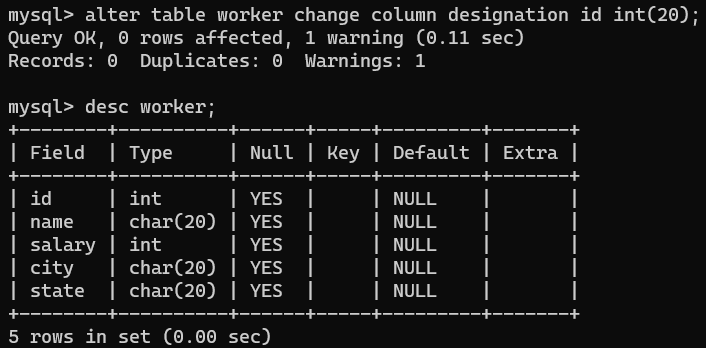
EX :-



1. Write a query to change column name.

Syntax :- alter table tablename change column columnname(old) columnname(new) datatype(size);

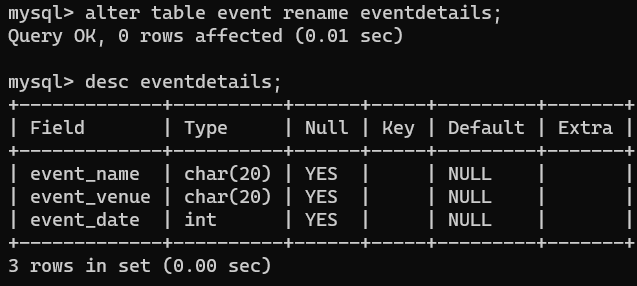
EX :-



1. Write a query to rename table event into eventdetails.

Syntax :- alter table tablename(old) rename tablename(new);

EX :-

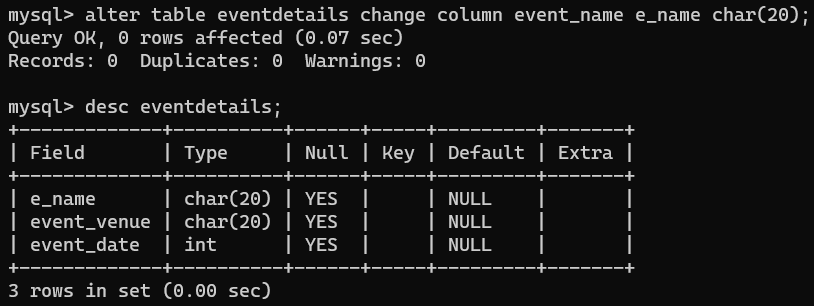


1. Write a query to rename column event\_name into

e\_name.

Syntax :- alter table tablename change column columnname(old) columnname(new) datatype(size);

EX :-

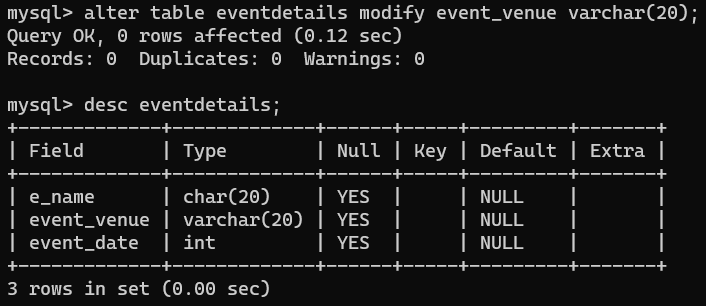


1. Write a query to change the datatype of column name

Event\_venue.

Syntax :- alter table tablename modify columnname datatype(size);

EX :-

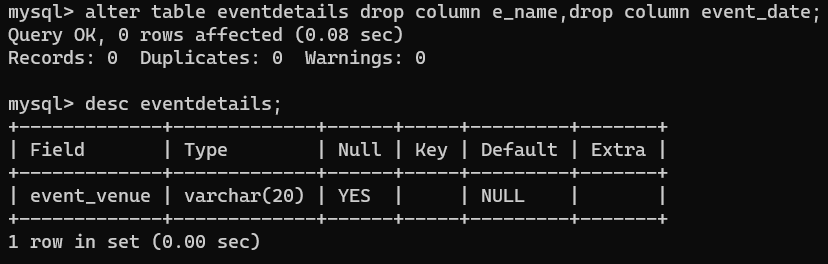


1. Write a query to drop column of e\_name and e\_date.

Syntax :-

1. Single :- alter table tablename drop column columnname;
2. Multiple :- alter table tablename drop column columnname1,drop column columnname2;

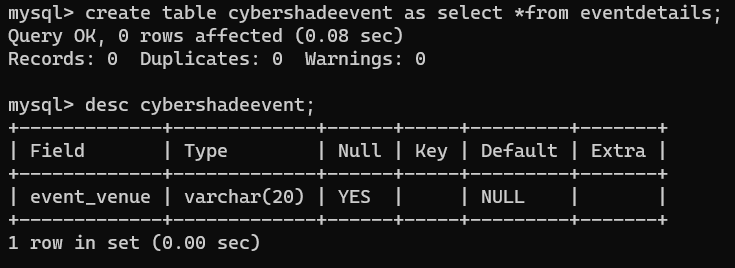
EX :-



1. Write a query to create a new table name cybershadeevent from an existing table name event with all the fields.

Syntax :- create table tablename(new) as select \*from tablename(old);

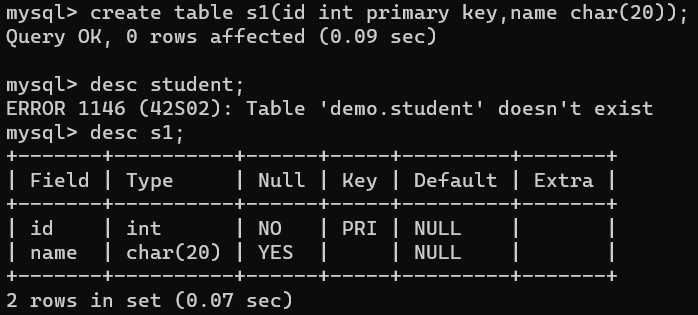
EX :-



1. Use of Primary Key (Only one exists in table).

Syntax :- create table tablename (columnname datatype primarykey , columnname2 datatype(size));

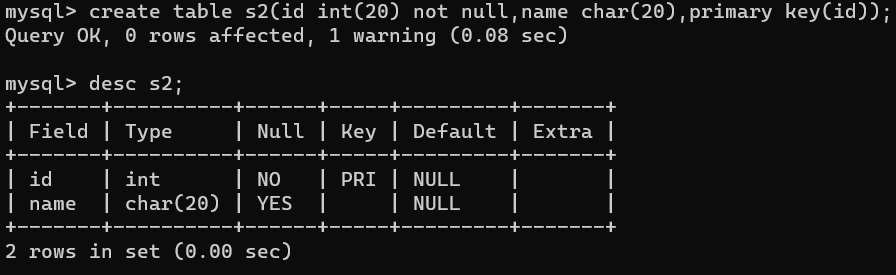
EX :-



1. Use of not null (to assign no) and Primary key at last.

Syntax :- create table tablename (columnname datatype(size) not null,columnname datatype(size),primary key(columnname);

EX :-

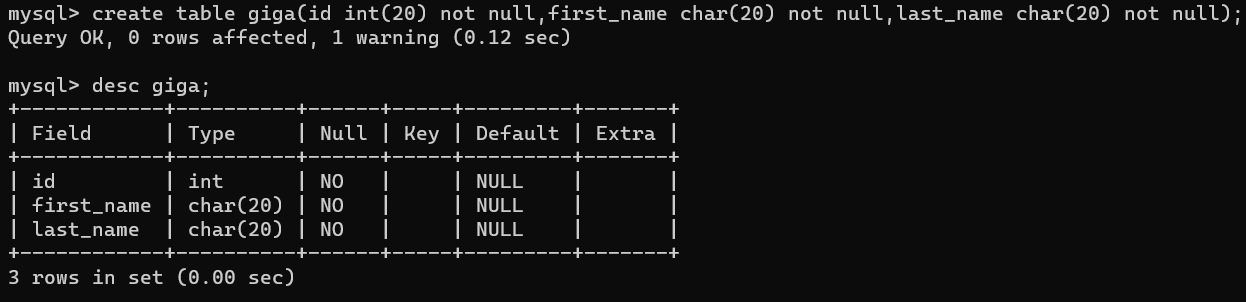


Note :- may be space is needed between primary and key.

1. Write a query that ensures column name id,first name,last name will not accept any null values when employe details table is created.

Syntax :- create table tablename (columnname datatype(size) not null,columnname datatype(size));

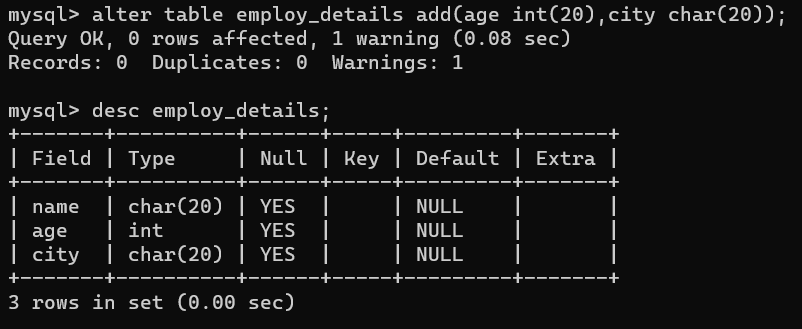
EX :-



1. Write a query to add column age and city to table employ\_details.

Syntax :- alter table tablename add(columnname1 datatype(size),columnname2 datatype(size));

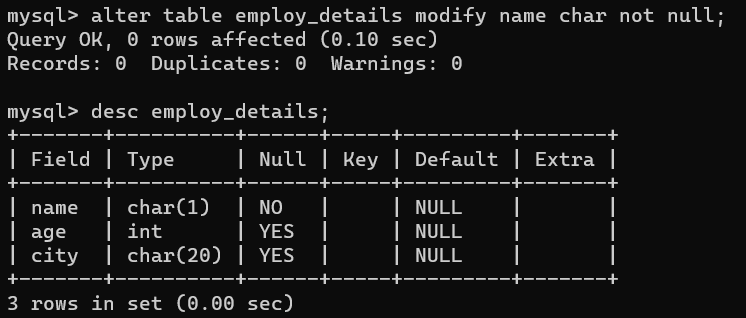
EX :-



1. Write a query to alter the constrain on columnname age where employ table is already created.

Syntax :- alter table tablename modify columnname datatype not null;

EX :-



1. Use of Insert value in table.

Syntax :- insert into tablename values(value1,value2);

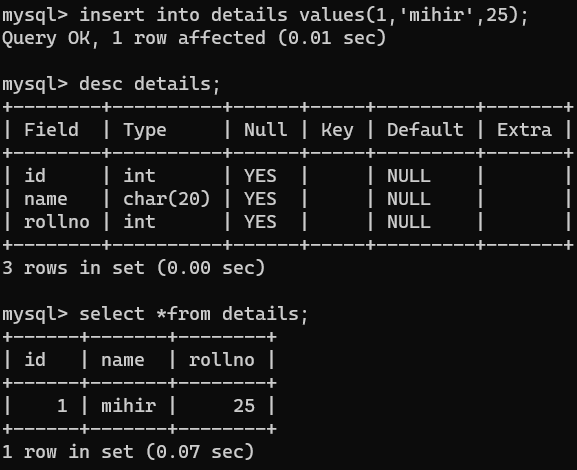
OR

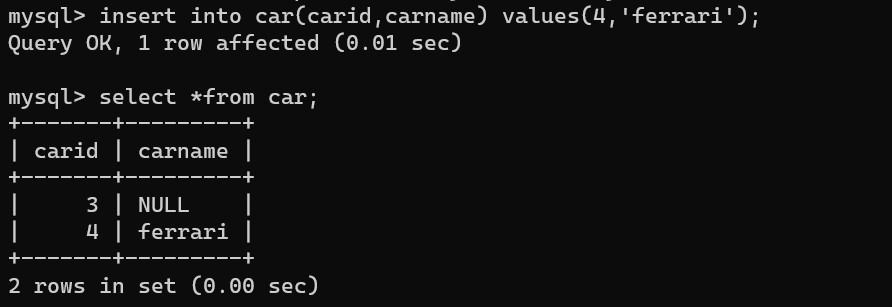
Syntax :- insert into tablename(column1,column2) values(value1,value2);

To show the insert value in table

Syntax :- select \*from tablename;

EX :-



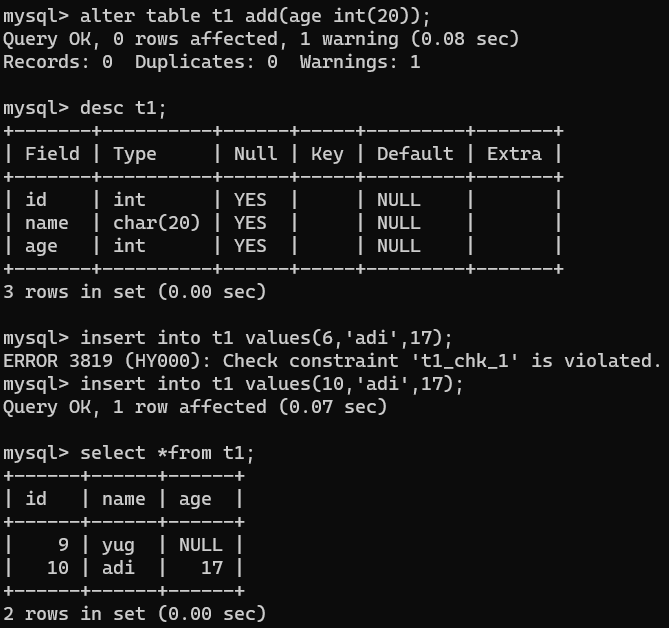


Note :- use ‘’ for character.

1. Write a query to alter the check constrain on age colum.

Syntax :- while creating table check(culumnname condition);

EX :-



1. Write a query to add multiple check constrain on multiple selected column.

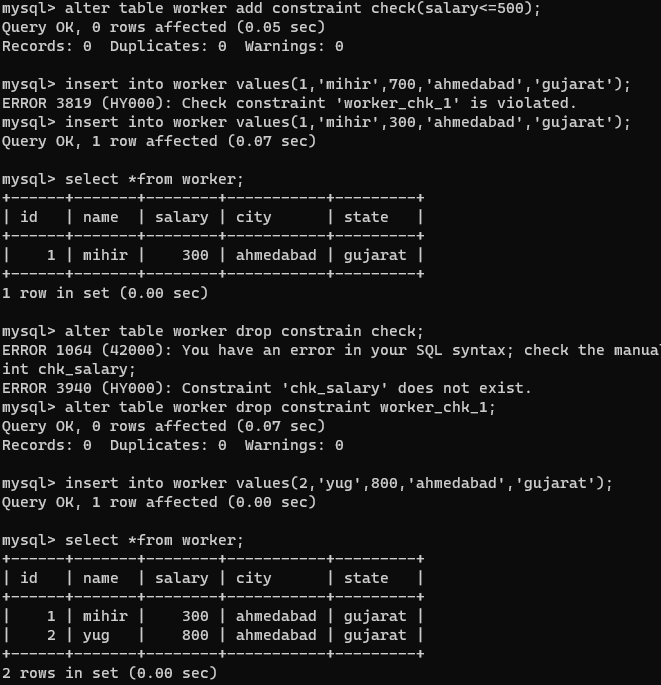
Syntax :- alter table tablename add constraint constraintname (columnname1 condition and columnname2 condition);

EX :-

1. To drop check constraint

Syntax :- alter table tablename drop constraint constraintname;

EX :-



1. To create foreign key

Syntax :- create table tablename (columnanme1 datatype(size),columnname2 datatype foreign key references tablename (columnname)

EX :-

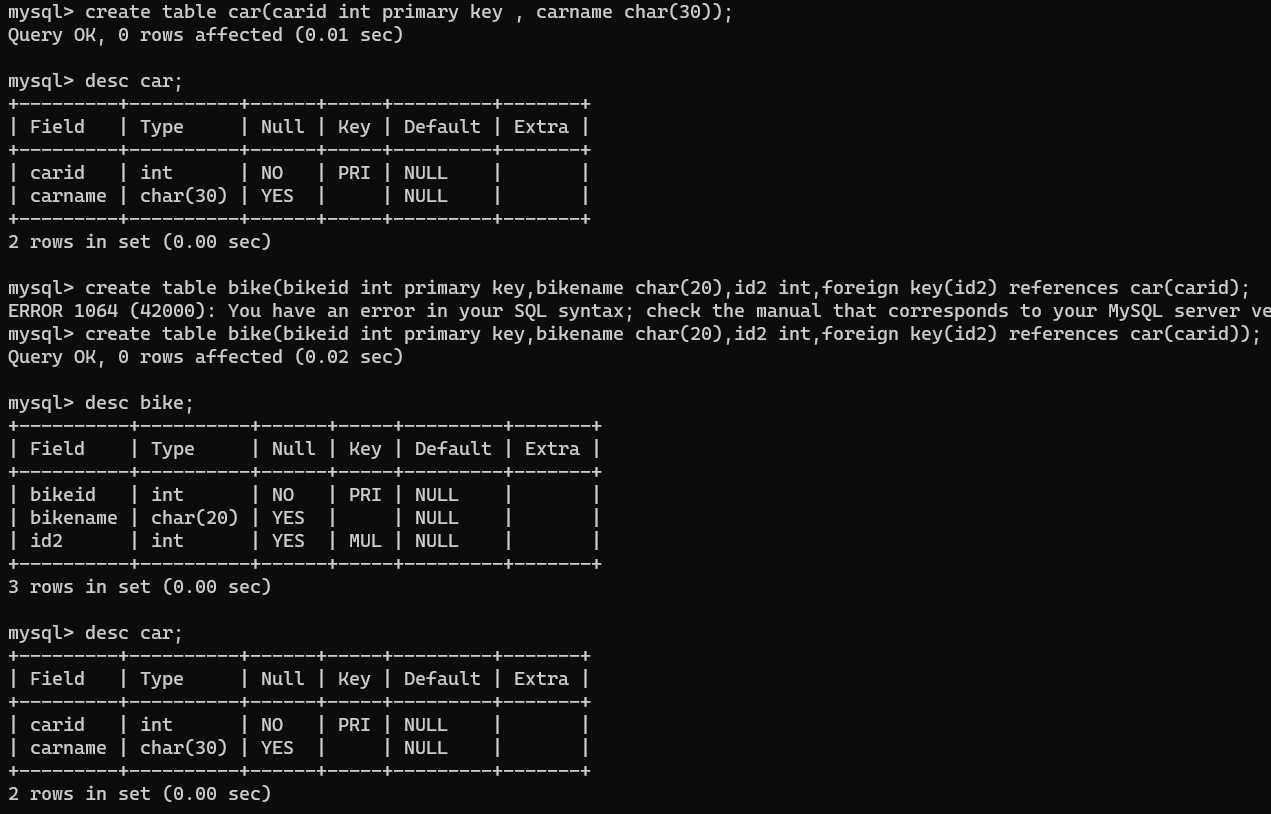
1. To make primary key

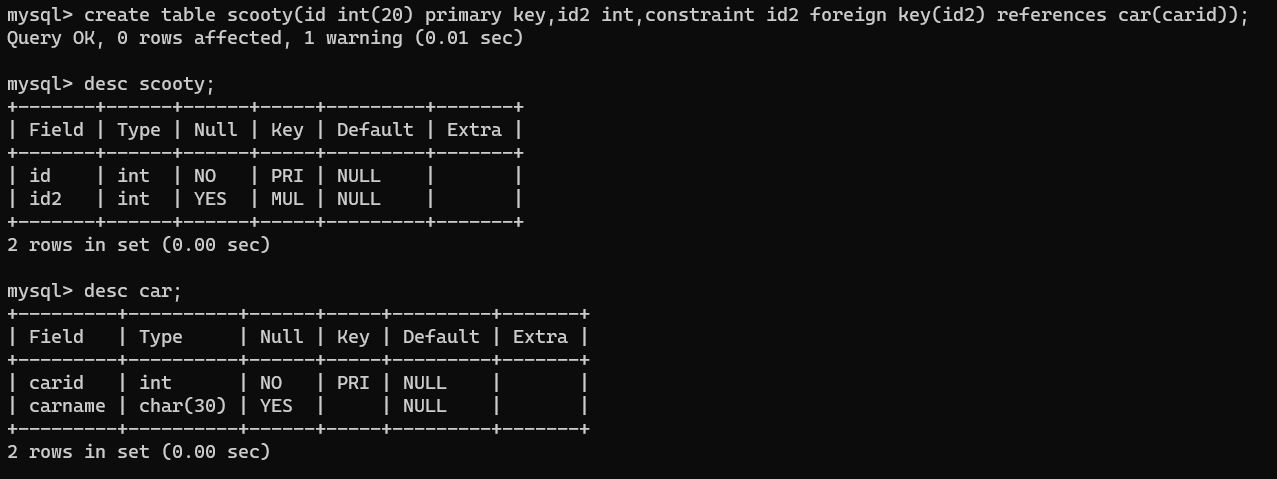
Syntax :- create table tablename (columnname1 datatype(size) primary key , columnname 2 datatype(size), constraint(id2) datatype,foreign key(constraint(id2)) references tablename(other table) (columnname{other table})

OR

create table tablename (columnname1 datatype(size) primary key , constraint constraintname foreign key(column name) references tablename(other table) (columnname{other table})

EX :-





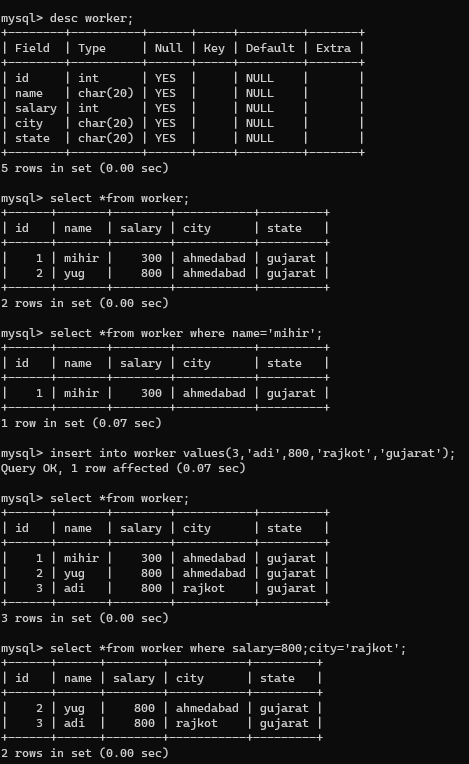
1. Retrive information based on condition

Syntax :-

Single :- select \*from tablename where condition

Multiple :- select \*from tablename where condition1;condition2;

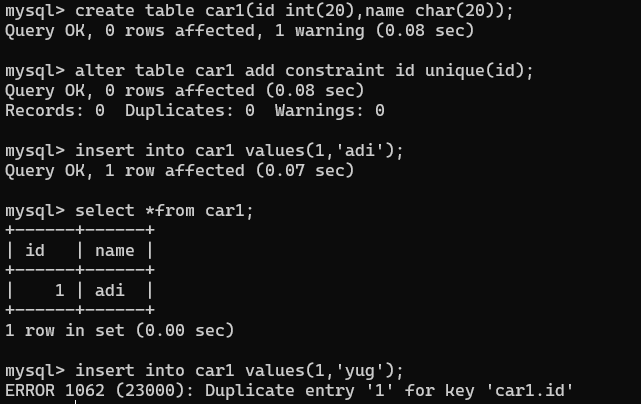
EX :-



1. To create unique number

Syntax :- alter table tablename add constraint constraintname unique (columnname)

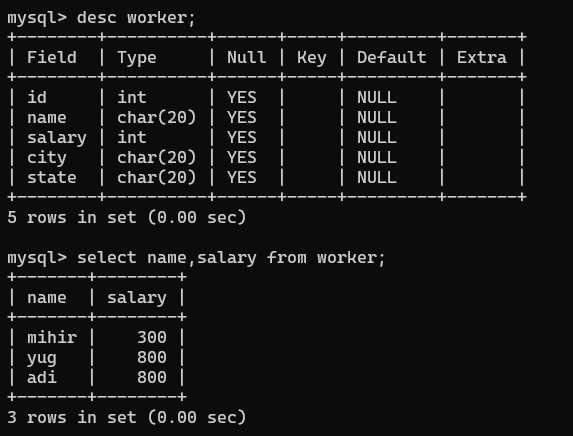
EX :-



1. To retrive specific column

Syntax :- select columnname1 , columnname 2 from tablenname;

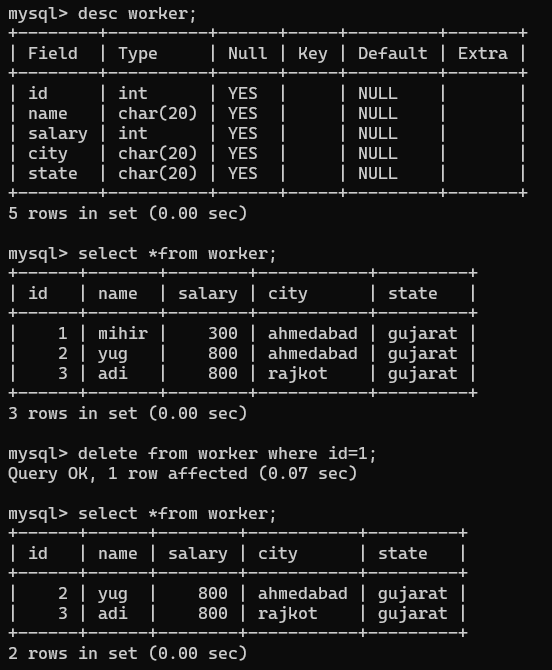
EX :-



1. Delete table with condition

Syntax :- delete from tablename where condition

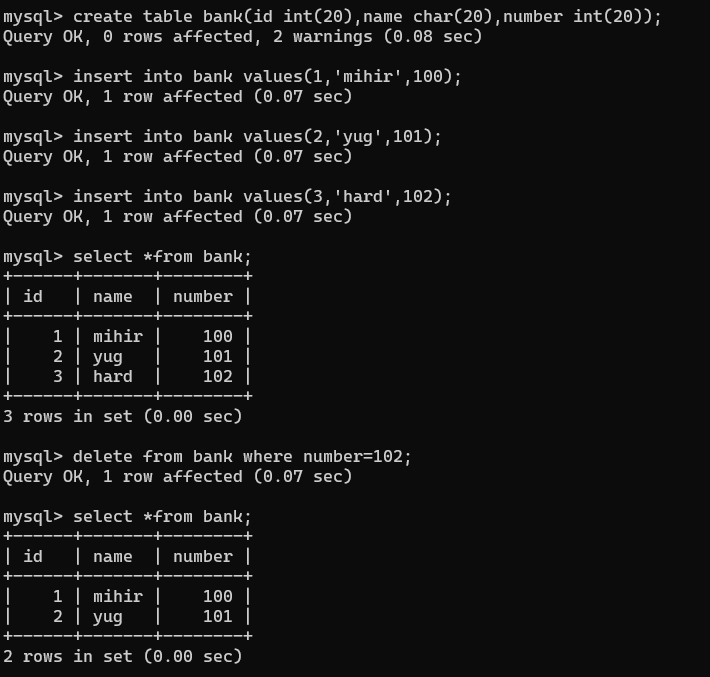
EX :-



1. Write a sql query to delete record from bank table whose account number is 102.

Syntax :- delete from tablename where condition

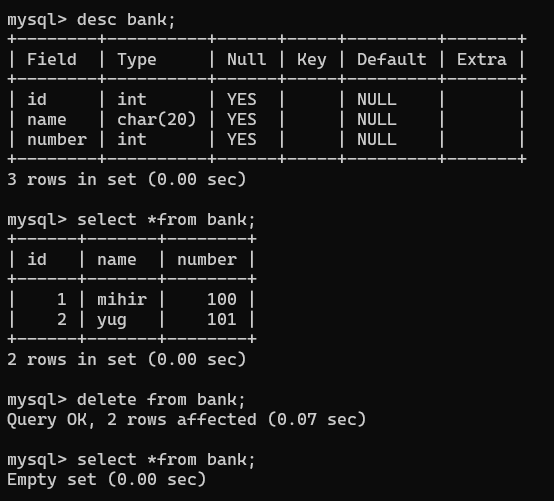
EX :-



1. To delete all the rows and columns from table (to empty inner table record)

Syntax :- delete from tablename;

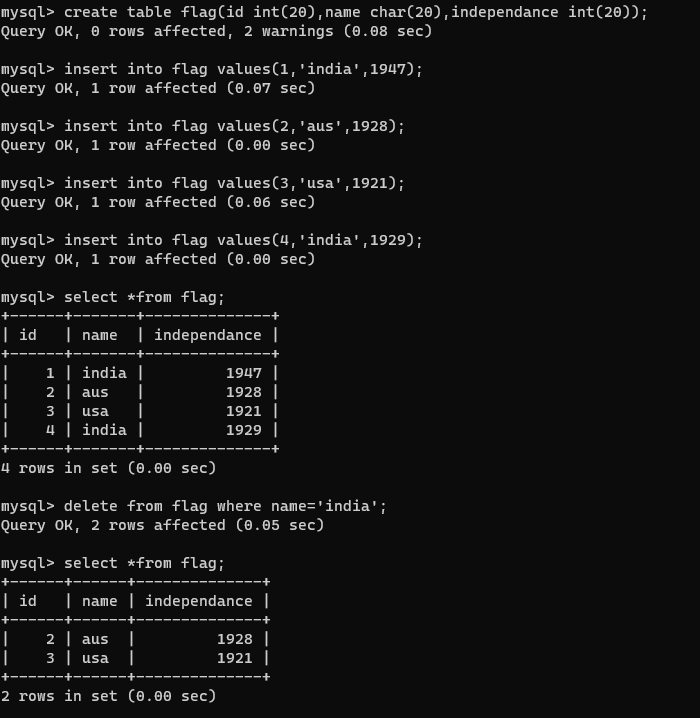
EX :-



1. Write a query to delete all the records from order table where country value is india.

Syntax :- delete from tablename where condition

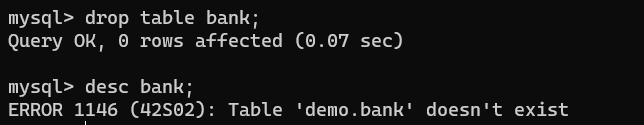
EX :-



1. To delete table along with records

Syntax :- Drop table tablename

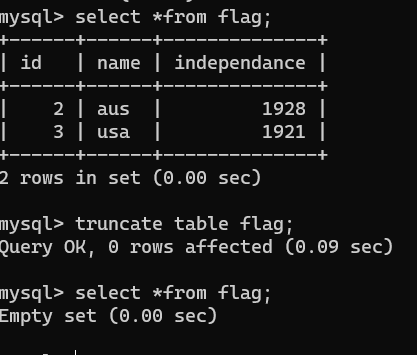
EX :-



1. To drop all table records (only)

Syntax :- truncate table tablename

Ex :-

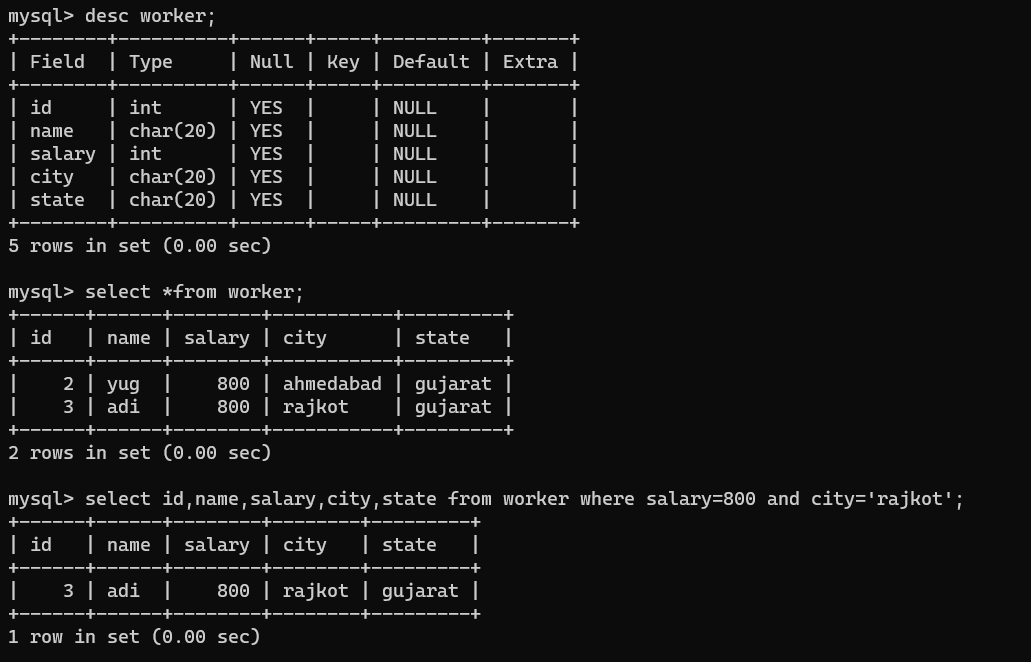


**Logical Operators**

1. AND :- two condition must be true then only print

Syntax :- select columnname1,columnname2(this column we want to show) from tablename where columname1 condition AND columnname2 condition

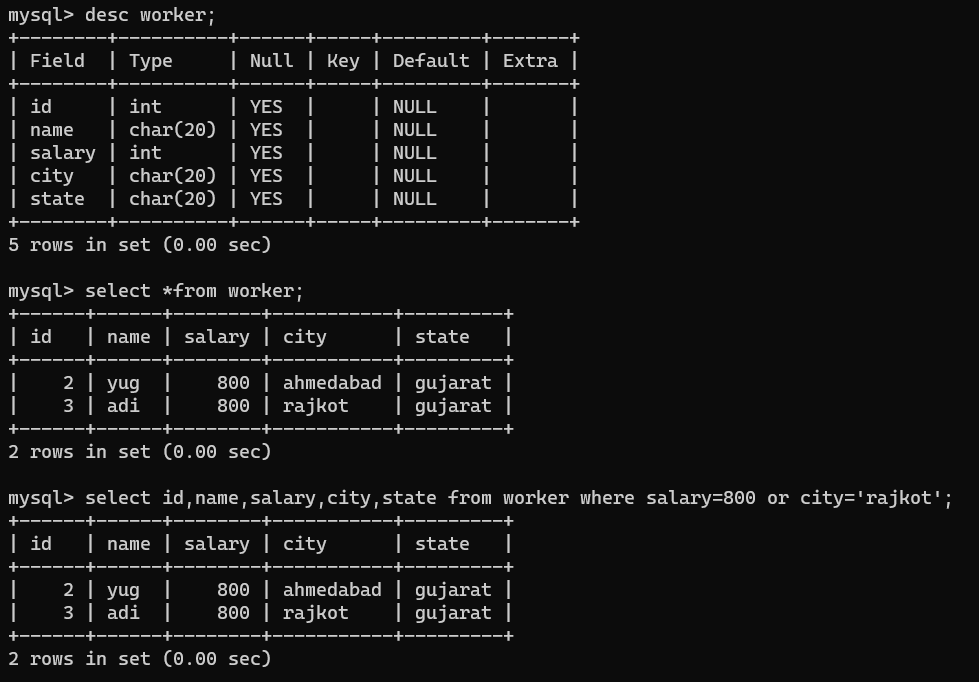
EX :-



1. OR :- if any one condition true print

Syntax :- select columnname1,columnname2(this column we want to show) from tablename where columname1 condition OR columnname2 condition

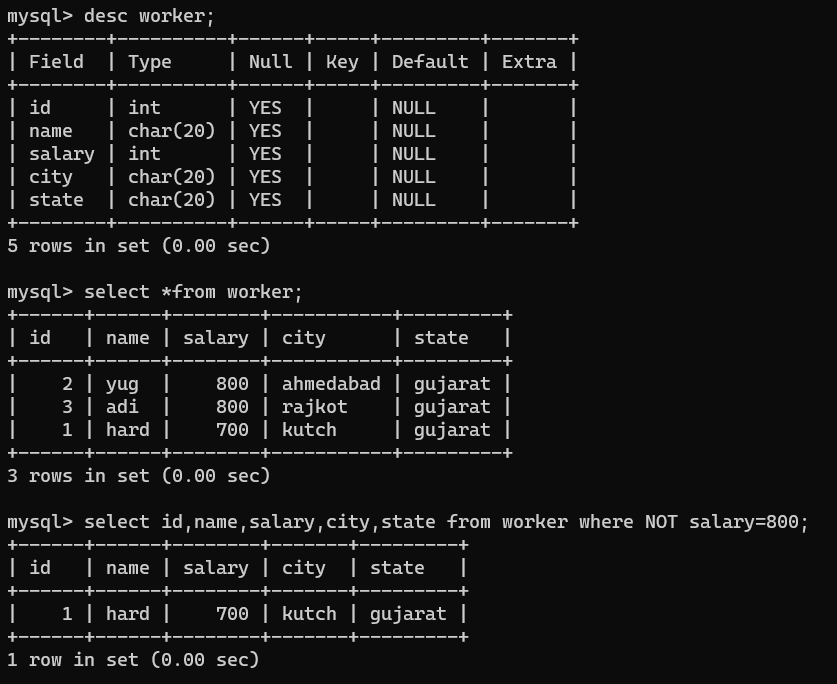
EX :-



1. NOT :- To print remain(not condition)

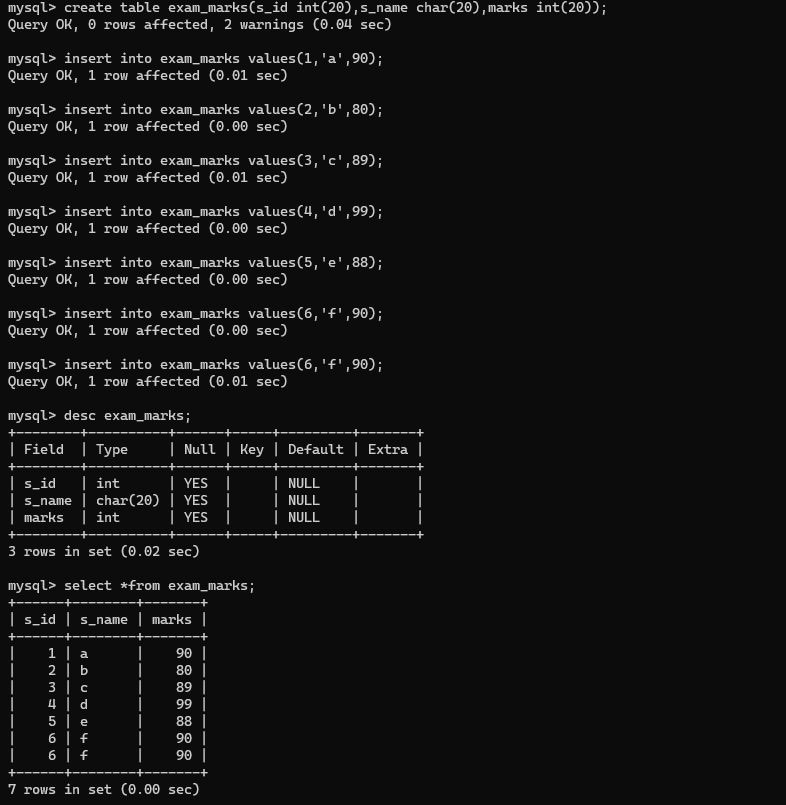
Syntax :- select columnname1,columnname2(this column we want to show) from tablename where NOT columnname condition

EX :-



**Aggregate Function**

Create table exam\_marks which contains s\_id , s\_name and marks column.

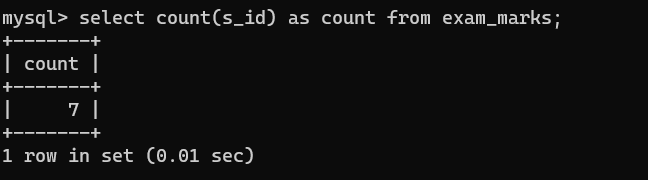


1. Count

Syntax :- select count(columnname) as count from tablename

Q – Find the total number of students.

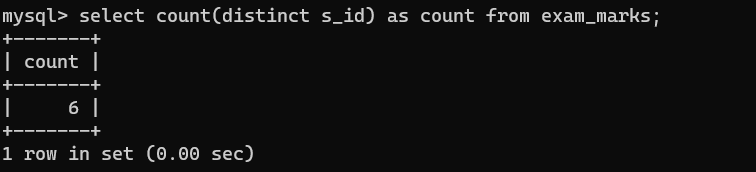
EX :-



* To remove duplicate in print

Syntax :- select count(distinct columnname) as count from tablename

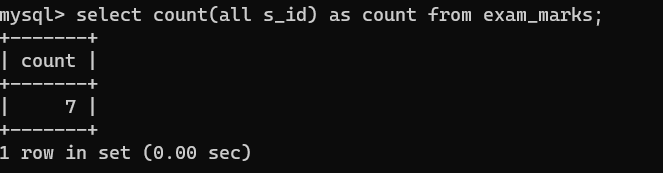
EX :-



* To show all along with duplicate

Syntax :- select count(all columnname) as count from tablename;

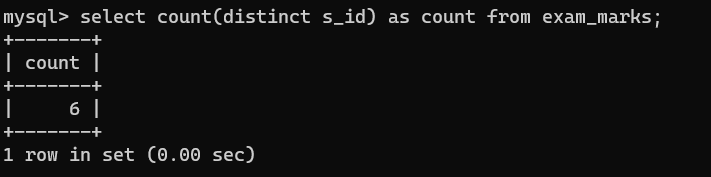
EX :-



Q – Find the total number of different students.

Syntax :- select count(distinct columnname) as count from tablename;

EX :-

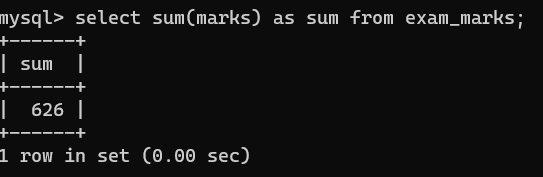


1. Sum

Syntax :- select sum(columnname) as sum from tablename

Q – Find total of marks scored by all students.

EX :-

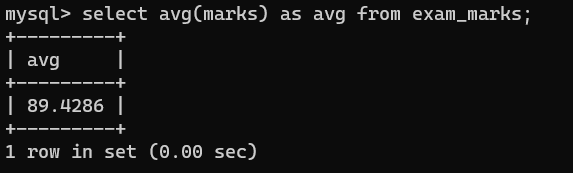


1. Avg

Syntax :- select avg(columnname) as avg from tablename;

Q – Find average marks of all student.

EX :-

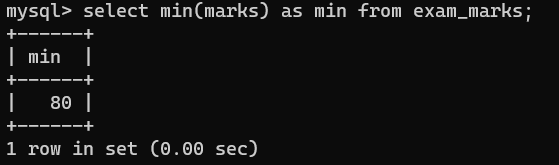


1. Min

Syntax :- select min(columnname) as min from tablename;

Q – Find minimum marks scored by students.

EX :-

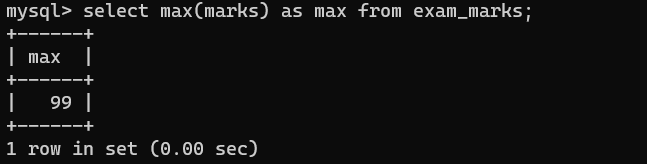


1. Max

Syntax :- select max(columnname) as max from tablename;

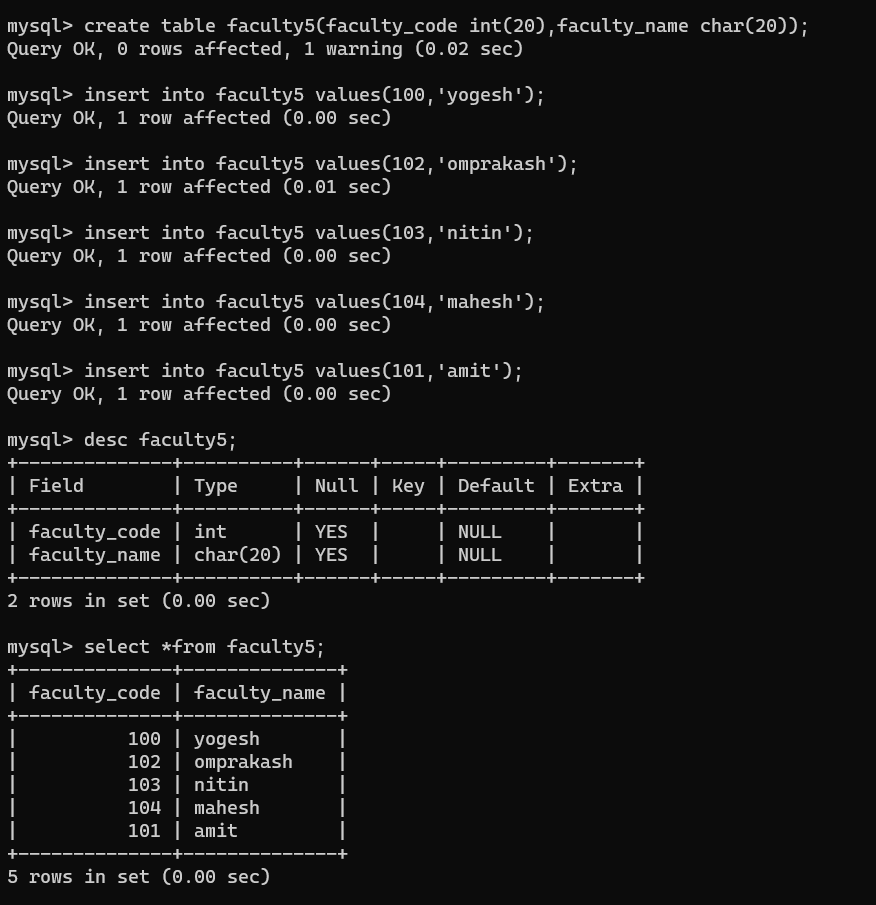
Q – Find maximum marks scored by students.

EX :-



**Pattern Matching String Operation**

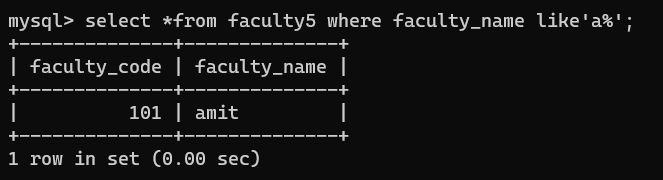
* SQL provides to wildcard characters for use with like.
* % - This sign is used to define start and end or string.
* \_ - This sign means a single character.
* Create table which contains faculty code and faculty name.



1. Find all the faculty name starts with a.

Syntax :- select \*from tablename where columnname like’a%’;

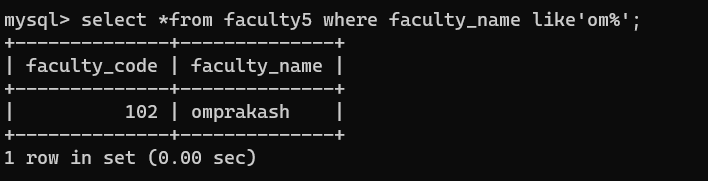
EX :-



1. Write a query to find all faculty’s whose name starts with om.

Syntax :- select \*from tablename where columnname like’om%’;

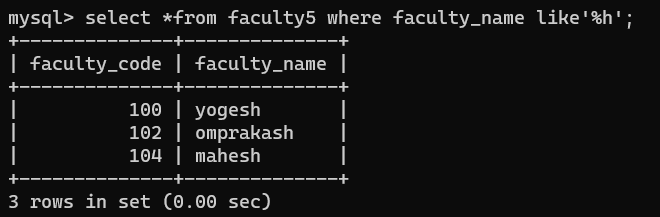
EX :-



1. Find all faculties whose name ends with h.

Syntax :- select \*from tablename where columnname like’%h’;

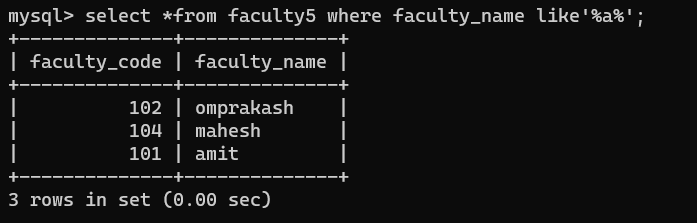
EX :-



1. Find all faculties whose name contains letter a.

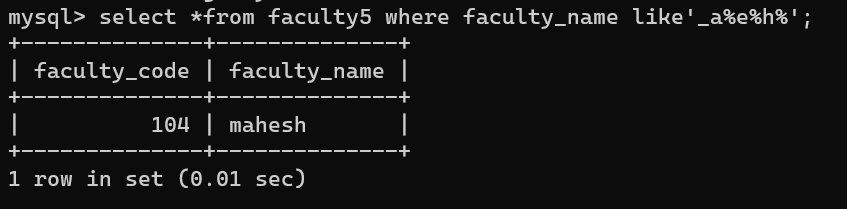
Syntax :- select \*from tablename where columnname like’%a%’;

EX :-



1. Find all faculties whose name second letter is a,contains letter e and name ending with letter h.

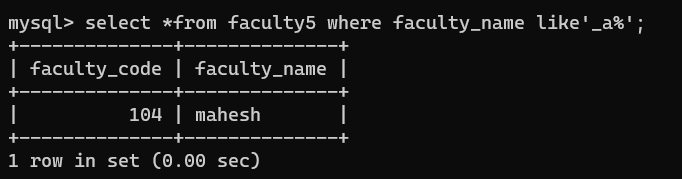
EX :-



1. Find all faculties whose names second letter is a.

Syntax :- select \*from tablename where columnname like’\_a%’;

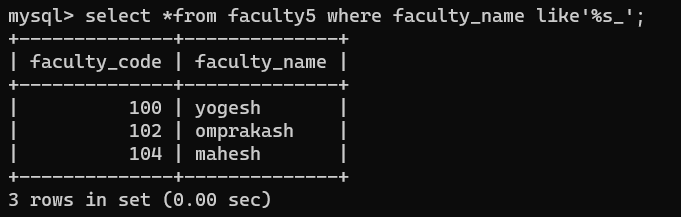
EX :-



1. Find all faculties whose name second last letter is s.

Syntax :- select \*from tablename where columnname like’%s\_’;

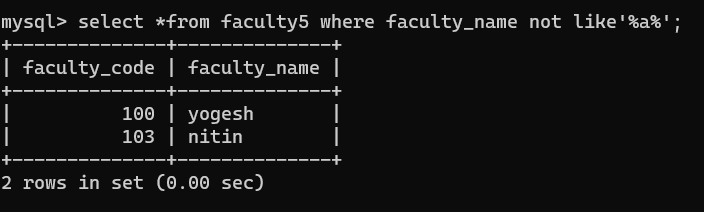
EX :-



1. Find all faculties whose name do not contains a.

Syntax :- select \*from tablename where columnname not like’%a%’;

EX :-

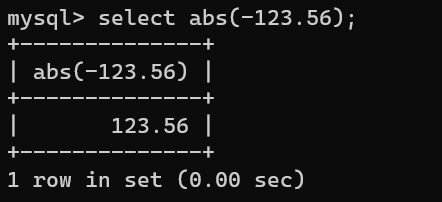


**Numeric Function**

1. ABS() :- Returns the absolute value of numeric expression.

Syntax :- Select abs(number);

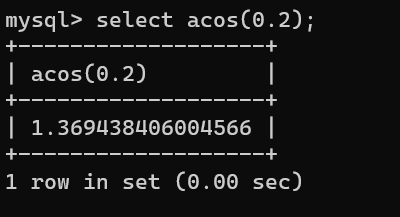
EX :-



1. ACOS() :- (inverse of cosine ) Returns the arcsine of numeric expression. Returns NULL if the value is not between the range -1 to 1.

Syntax :- select ACOS(number);

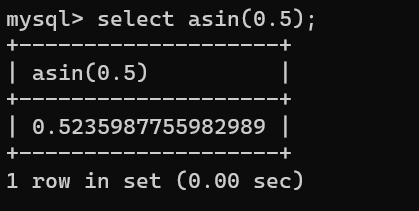
EX : -



1. ASIN() :- Return the arcsine of a number returns NULL if value is not the range between -1 to 1.

Syntax :- select ASIN(number);

EX :-



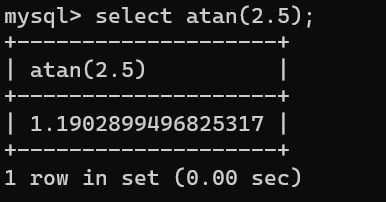
1. ATAN() :- Returns the arctangent of numeric expression.

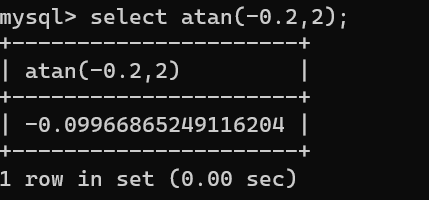
Syntax :- select ATAN(number);

OR

Select ATAN(a{num1},b{num2});

EX :-

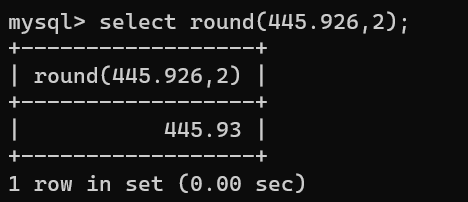




1. ROUND() :- Returns the value to specified decimal

Syntax :- select Round(a{num1},b{num2});

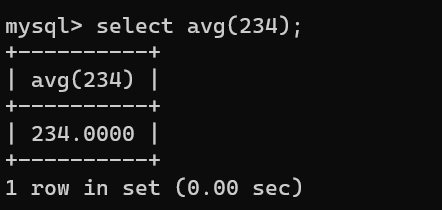
EX :-



1. AVG() :-Returns the average

Syntax :- select AVG(number);

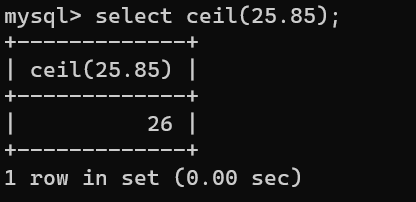
EX :-



1. CEIL() :- This function returns the smallest ineteger value that is not less than passed numeric expression.

Syntax :- select CEIL(number);

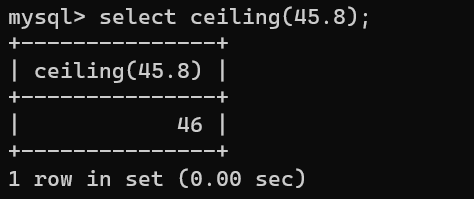
EX :-



1. CEILING() :- same functioning as CEIL()

Syntax :- select from CEILING(number);

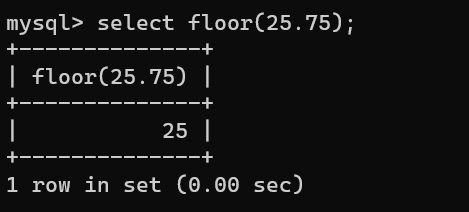
EX :-



1. FLOOR() :- Returns the largest integer value that is not greater than the passed numeric function.

Syntax :- select FLOOR(number);

EX : -

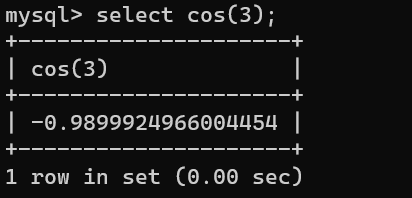


10)

COS() :- Return the cosine of a number. The numeric expression should be expressed in radians.

Syntax :- select COS(number);

EX :-

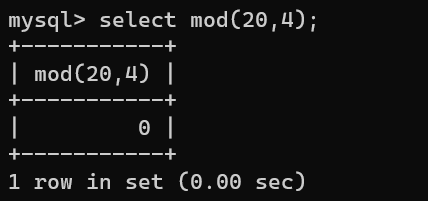


1. MOD() :- Returns the remainder of one expression by dividing by another expression.

Syntax :- select MOD(a{number},b{number});

A = number , B = divide

EX :-



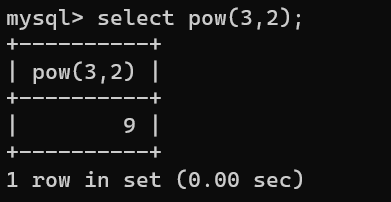
1. POW() :- Returns the value of one expression raised to the power of another expression.

Syntax :- select POW(num1,num2);

Num1 = number

Num2 = raise to

EX :-



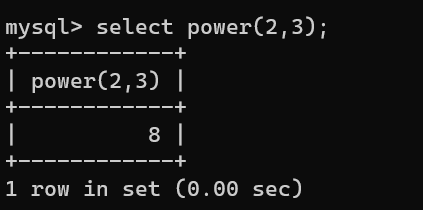
1. POWER() :- same as POW()

Syntax :- select POWER(num1,num2);

Num1 = number

Num2 = raise to

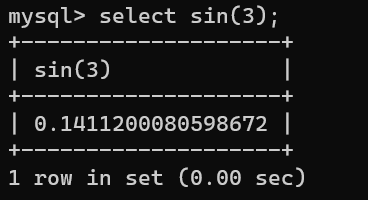
EX :-



1. SIN() :- Return the Sine of number

Syntax :- select SIN(number);

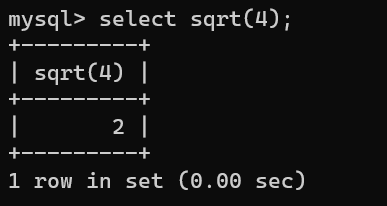
EX :-



1. SQRT() :- Return the square-root of number

Syntax :- select SQRT(number);

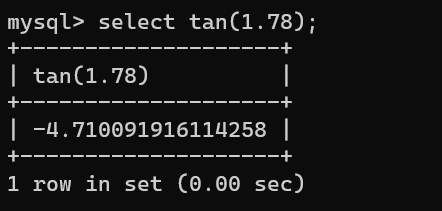
EX :-



1. TAN() :- Returns the tangent of a number

Syntax :- select TAN(number);

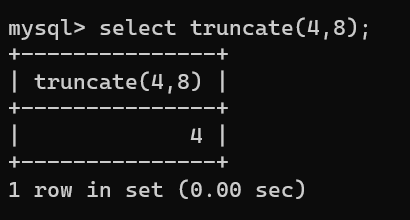
EX :-



1. TRUNCATE() :- Returns a number truncated by given decimal places.

Syntax :- select truncate(number,decimal number);

EX :-



1. LOG() :- Returns logarithm of given number

Syntax :- Select LOG(number);

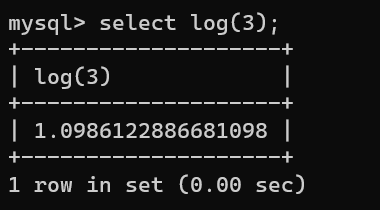
OR

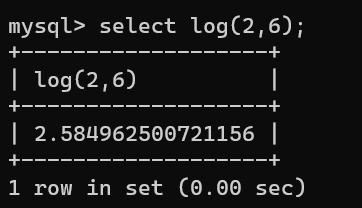
Select LOG(base,number)

Base = >1

Number = >0

EX :-



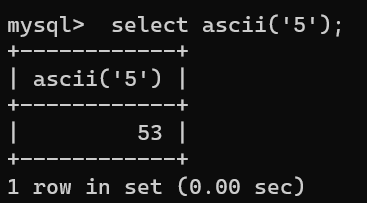


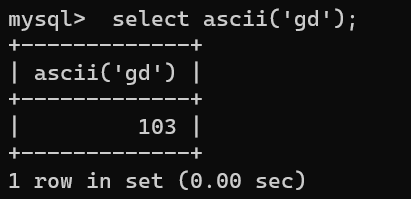
**String Functions**

1. ASCII(str)

Returns the numeric value of the leftmost character of the string str. Returns 0 if str is the empty string. Returns NULL if str is NULL. ASCII() works for characters with numeric values from 0 to 255.

EX :-

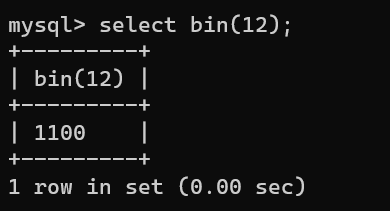




1. BIN(N)

Returns a string representation of the binary value of N, where N is a longlong (BIGINT) nuber. This is equivalent to CONV(N,10,2). Returns NULL if N is NULL.

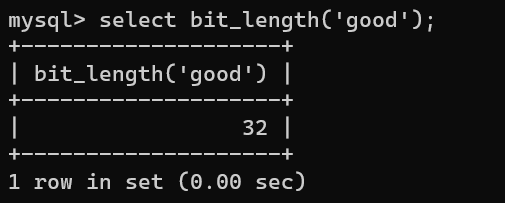
EX :-



1. BIT\_LENGTH(str)

Returns the length of the string str in bits.

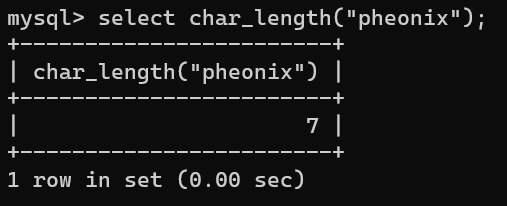
EX :-



1. CHAR\_LENGTH(str)

Returns the length of the string str, measured in characters. A multi-byte character counts as a single character. This means that for a string containing five two-byte characters, LENGTH() returns 10, whereas CHAR\_LENGTH() returns 5.

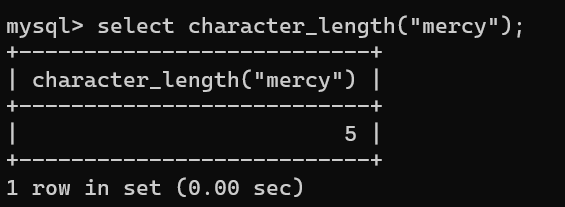
EX :-



1. CHARACTER\_LENGTH(str)

CHARACTER\_LENGTH() is a synonym for CHAR\_LENGTH().

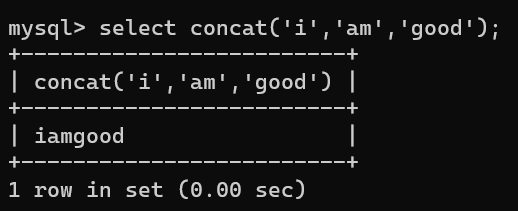
EX :-



1. CONCAT(str1,str2,...)

Returns the string that results from concatenating the arguments. May have one or more arguments. If all arguments are non-binary strings, the result is a non-binary string. If the arguments include any binary strings, the result is a binary string. A numeric argument is converted to its equivalent binary string form; if you want to avoid that, you can use an explicit type cast, as in this example

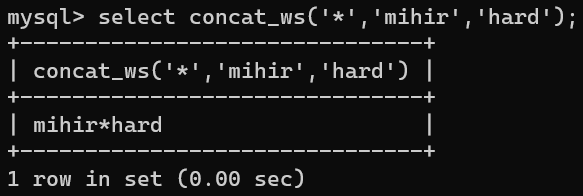
EX :-



1. CONCAT\_WS(separator,str1,str2,...)

CONCAT\_WS() stands for Concatenate With Separator and is a special form of CONCAT(). The first argument is the separator for the rest of the arguments. The separator is added between the strings to be concatenated. The separator can be a string, as can the rest of the arguments. If the separator is NULL, the result is NULL.

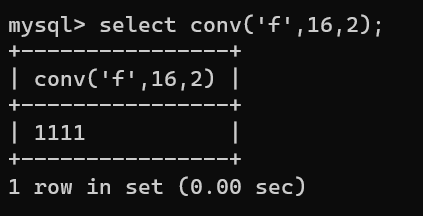
EX :-

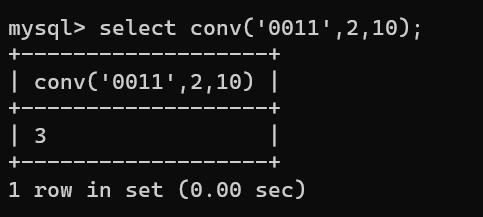


1. CONV(N,from\_base,to\_base)

Converts numbers between different number bases. Returns a string representation of the number N, converted from base from\_base to base to\_base. Returns NULL if any argument is NULL. The argument N is interpreted as an integer, but may be specified as an integer or a string. The minimum base is 2 and the maximum base is 36.

EX :-

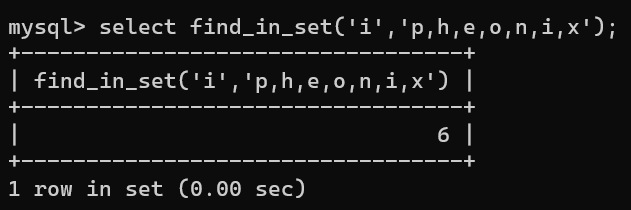




1. FIND\_IN\_SET(str,strlist)

Returns a value in the range of 1 to N if the string str is in the string list strlist consisting of N substrings.

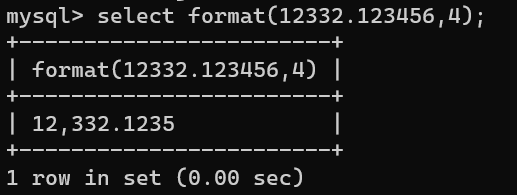
EX :-



1. FORMAT(X,D)

Formats the number X to a format like '#,###,###.##' , rounded to D decimal places, and returns the result as a string. If D is 0, the result has no decimal point or fractional part.

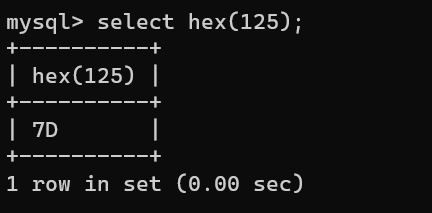
EX :-

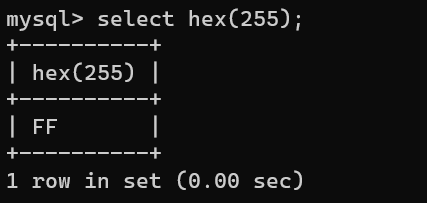


1. HEX(N\_or\_S)

If N\_or\_S is a number, returns a string representation of the hexadecimal value of N.

EX :-

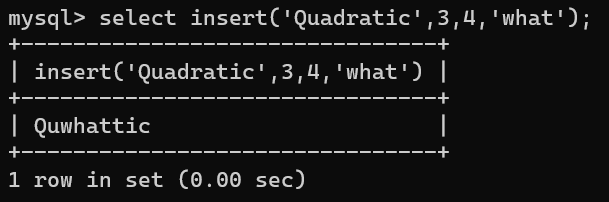




1. INSERT(str,pos,len,newstr)

Returns the string str, with the substring beginning at position pos and len characters long replaced by the string newstr. Returns the original string if pos is not within the length of the string. Replaces the rest of the string from position pos if len is not within the length of the rest of the string. Returns NULL if any argument is NULL.

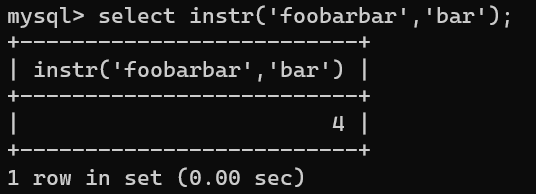
EX :-

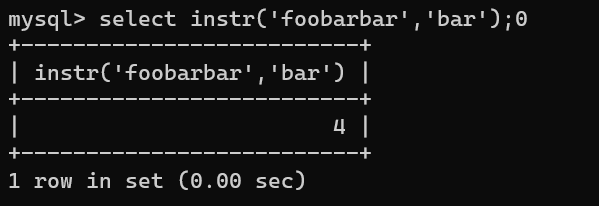


1. INSTR(str,substr)

Returns the position of the first occurrence of substring substr in string str. This is the same as the two-argument form of LOCATE(), except that the order of the arguments is reversed.

EX :-

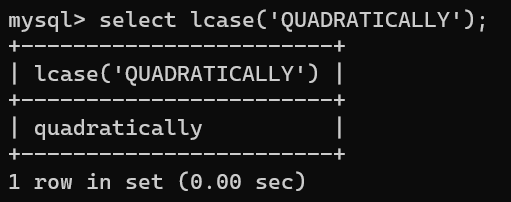




1. LCASE(str)

LCASE() is a synonym for LOWER().

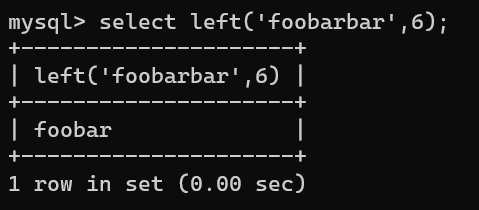
EX :-



1. LEFT(str,len)

Returns the leftmost len characters from the string str, or NULL if any argument is NULL.

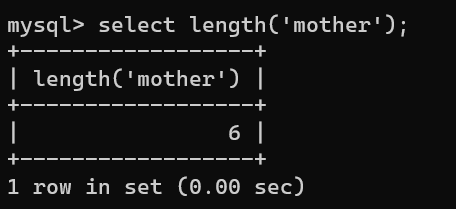
EX :-



1. LENGTH(str)

Returns the length of the string str, measured in bytes. A multi-byte character counts as multiple bytes. This means that for a string containing five two-byte characters, LENGTH() returns 10, whereas CHAR\_LENGTH() returns 5.

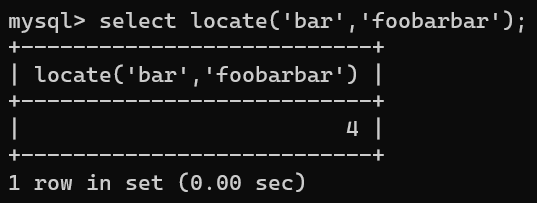
EX :-



1. LOCATE(substr,str), LOCATE(substr,str,pos)

The first syntax returns the position of the first occurrence of substring substr in string str. The second syntax returns the position of the first occurrence of substring substr in string str, starting at position pos. Returns 0 if substr is not in str.

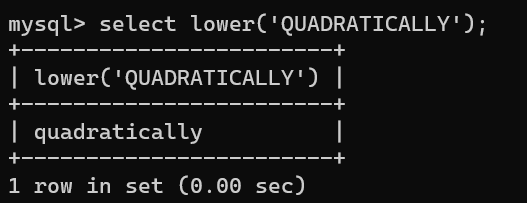
EX :-



1. LOWER(str)

Returns the string str with all characters changed to lowercase according to the current character set mapping.

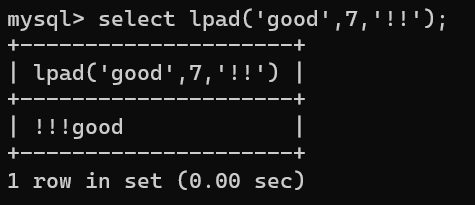
EX :-



1. LPAD(str,len,padstr)

Returns the string str, left-padded with the string padstr to a length of len characters. If str is longer than len, the return value is shortened to len characters.

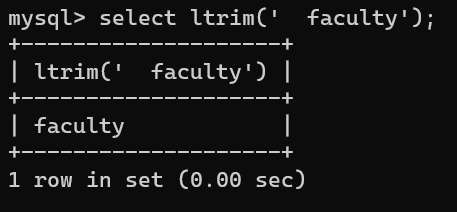
EX :-



1. LTRIM(str)

Returns the string str with leading space characters removed.

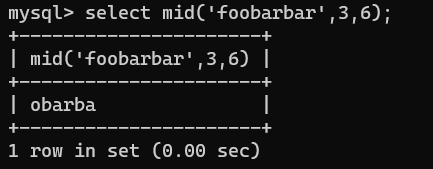
EX :-



1. MID(str,pos,len)

MID(str,pos,len) is a synonym for SUBSTRING(str,pos,len).

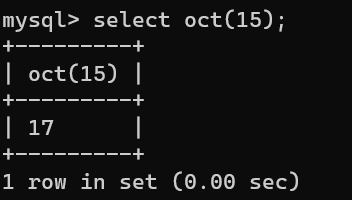
EX :-



1. OCT(N)

Returns a string representation of the octal value of N, where N is a longlong (BIGINT) number.

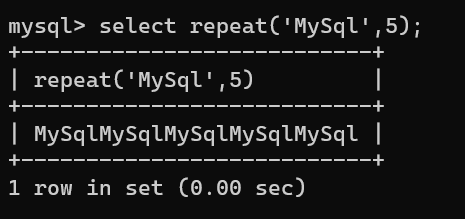
EX :-



1. REPEAT(str,count)

Returns a string consisting of the string str repeated count times. If count is less than 1, returns an empty string. Returns NULL if str or count are NULL.

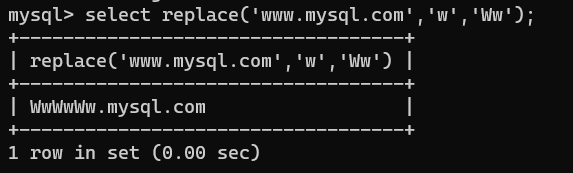
EX :-



1. REPLACE(str,from\_str,to\_str)

Returns the string str with all occurrences of the string from\_str replaced by the string to\_str. REPLACE() performs a case-sensitive match when searching for from\_str.

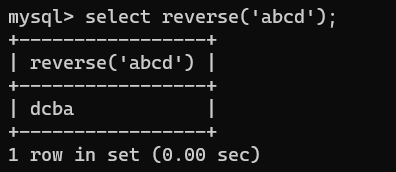
EX :-



1. REVERSE(str)

Returns the string str with the order of the characters reversed.

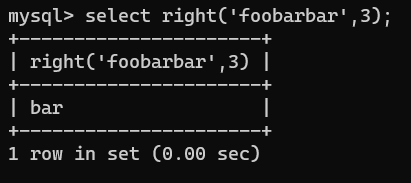
EX :-



1. RIGHT(str,len)

Returns the rightmost len characters from the string str, or NULL if any argument is NULL.

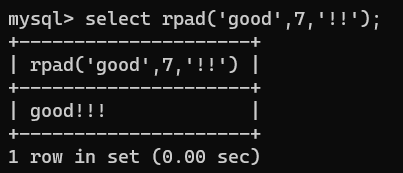
EX :-



1. RPAD(str,len,padstr)

Returns the string str, right-padded with the string padstr to a length of len characters. If str is longer than len, the return value is shortened to len characters.

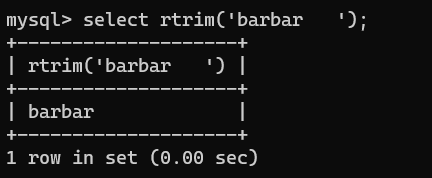
EX :-



1. RTRIM(str)

Returns the string str with trailing space characters removed.

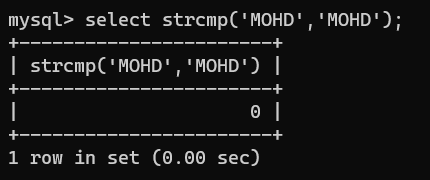
EX :-

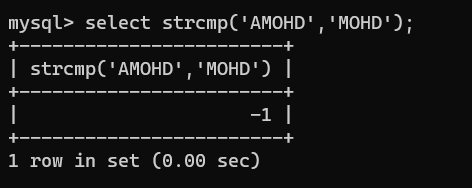


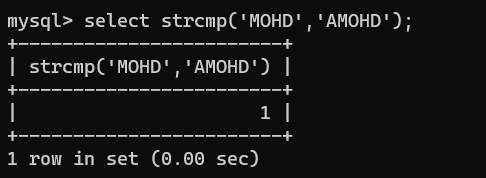
1. STRCMP(str1, str2)

Compares two strings and returns 0 if both strings are equal, it returns -1 if the first argument is smaller than the second according to the current sort order otherwise it returns 1.

EX :-







1. SUBSTRING(str,pos)

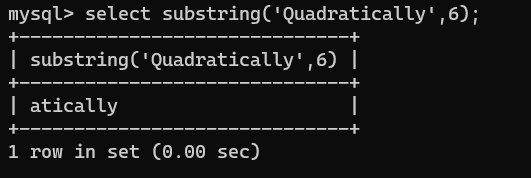
SUBSTRING(str FROM pos)

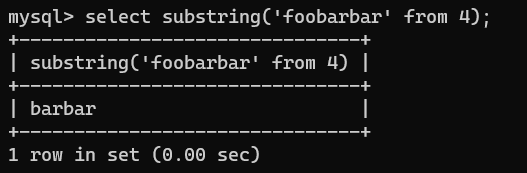
SUBSTRING(str,pos,len)

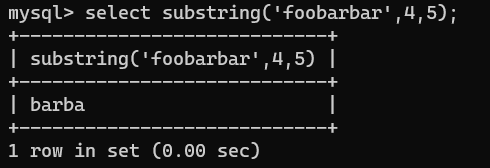
SUBSTRING(str FROM pos FOR len)

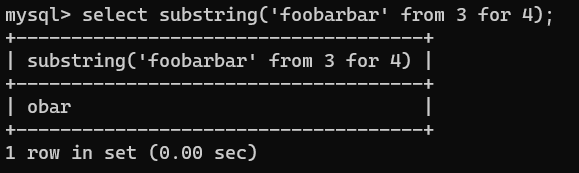
The forms without a len argument return a substring from string str starting at position pos. The forms with a len argument return a substring len characters long from string str, starting at position pos. The forms that use FROM are standard SQL syntax. It is also possible to use a negative value for pos. In this case, the beginning of the substring is pos characters from the end of the string, rather than the beginning. A negative value may be used for pos in any of the forms of this function.

EX :-



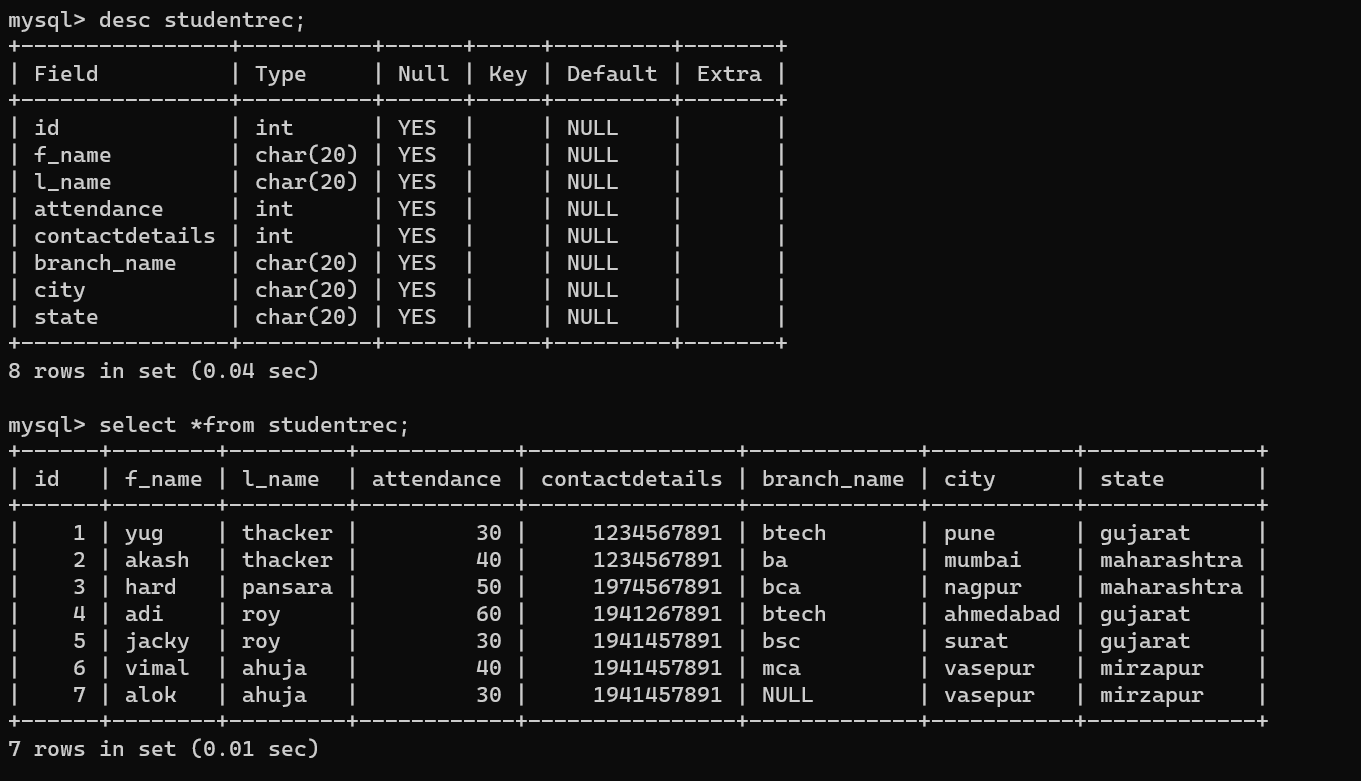






**More Queries**

* Create a studentrecord table which contains id , firstname , lastname , attendance , contactdetails , branchname , city , state.



**IN AND NOT IN**

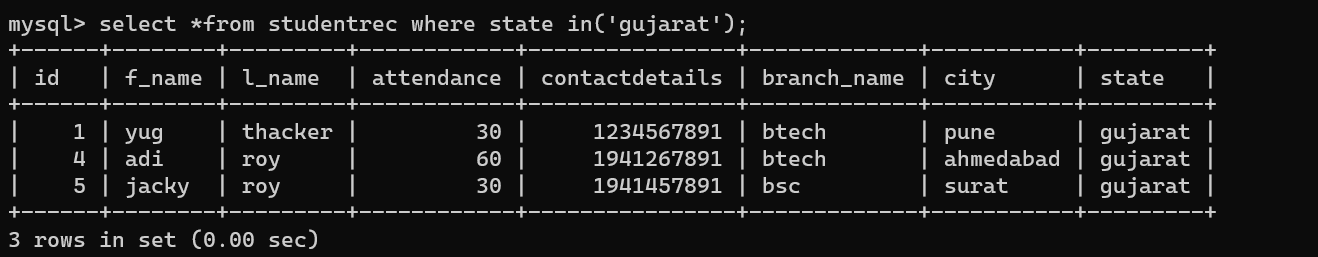
1. Write a query to display student record who belongs to Gujarat state.

Syntax :- select \*from tablename where columnname IN(value1,value2);

OR

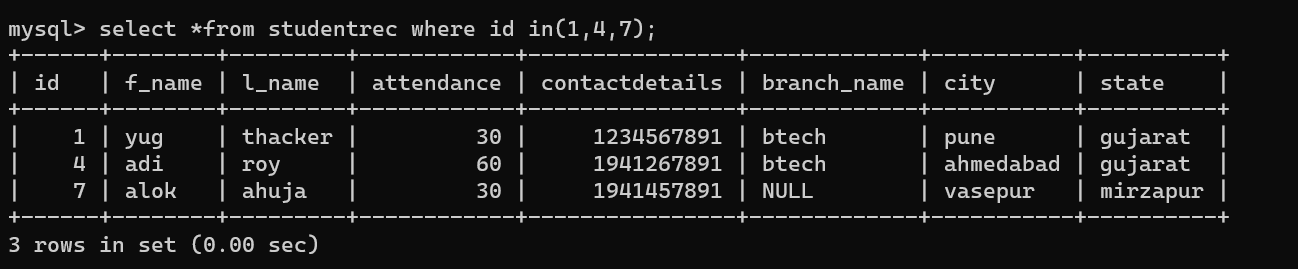
select \*from tablename where columnname NOT IN(value1,value2);

EX :-



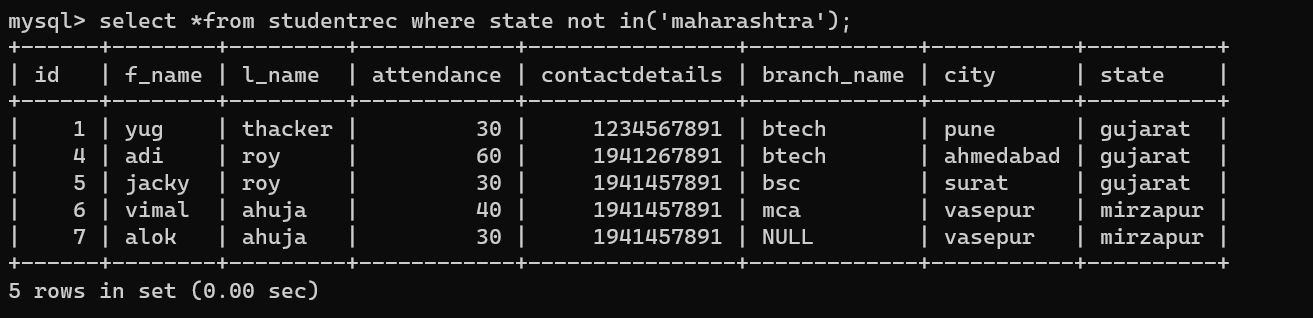
1. Write a query to display student record whose student id is 1,4,7.

EX :-



1. Write a query to display student record who doesnot belong to Maharashtra.

EX :-



**Between and Not Between**

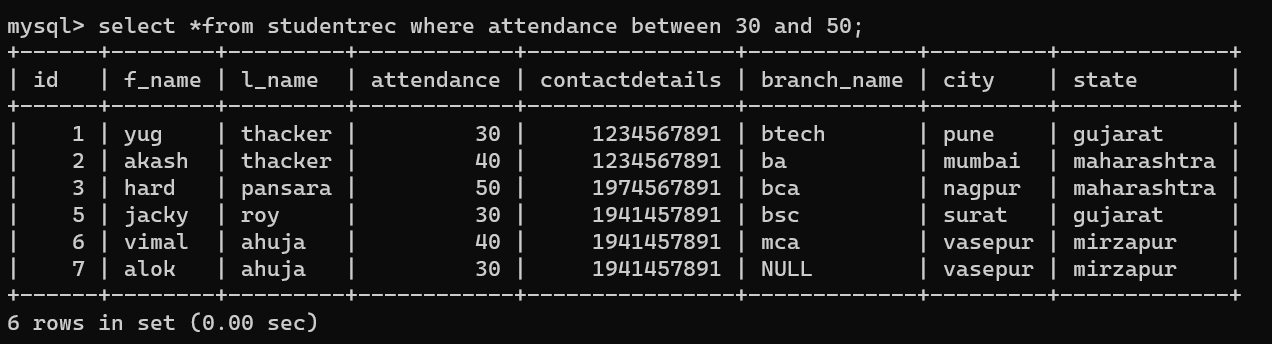
Syntax :- select \*from tablename where columnname between 30 and 40;

OR

select \*from tablename where columnname not between 30 and 40;

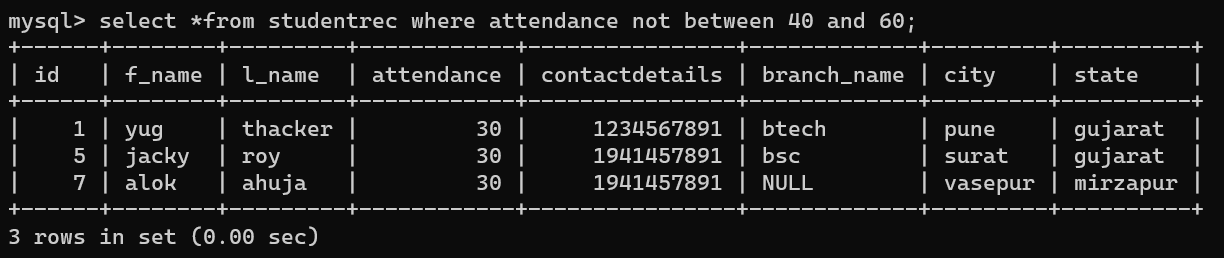
1. Write a query to display student records whose attendance is between 30 t0 50.

EX :-



1. Write a query to display student records whose attendance is not between 40 to 60.

EX :-

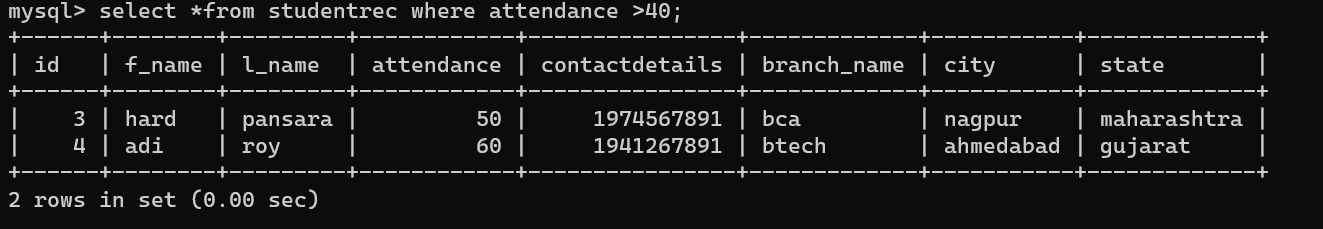


**Comparison Operator**

Syntax :- select \*from tablename where operator(condition);

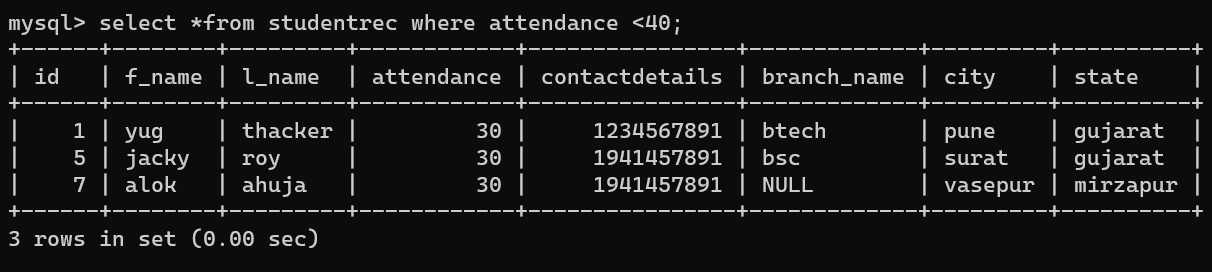
1. Write a query to display student record whose student attendance is more than 40.

EX :-



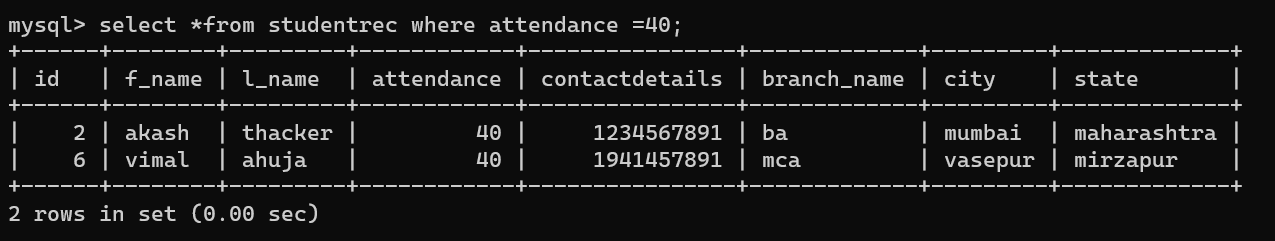
1. Write a query to display student record whose student attendance is less than 40.

EX :-



1. Write a query to display student record whose student attendance is 50.

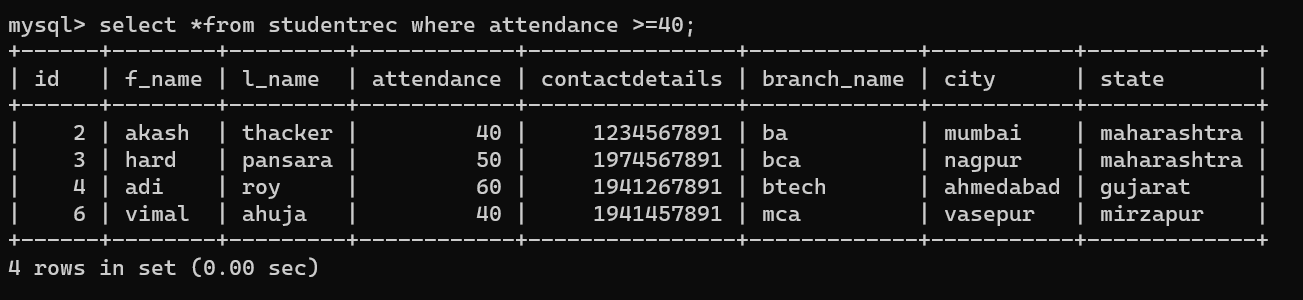
EX :-



1. Write a query to display student record whose student

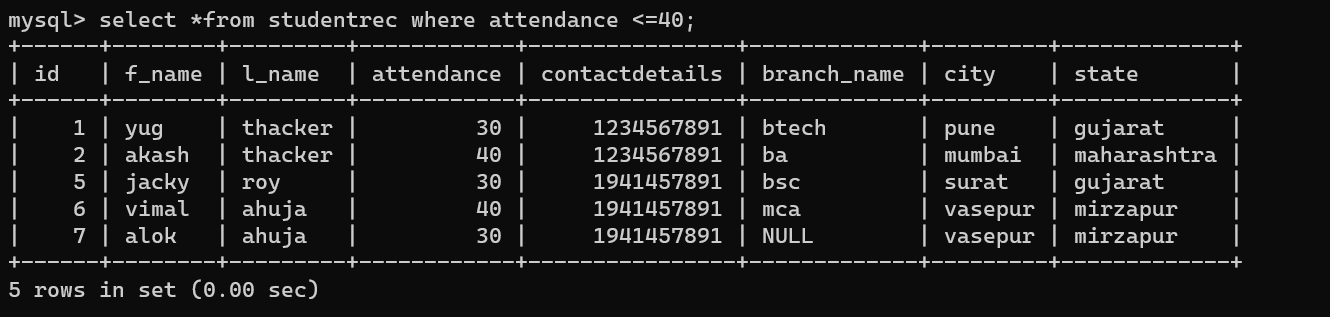
Attendance is 40 or more.

EX :-



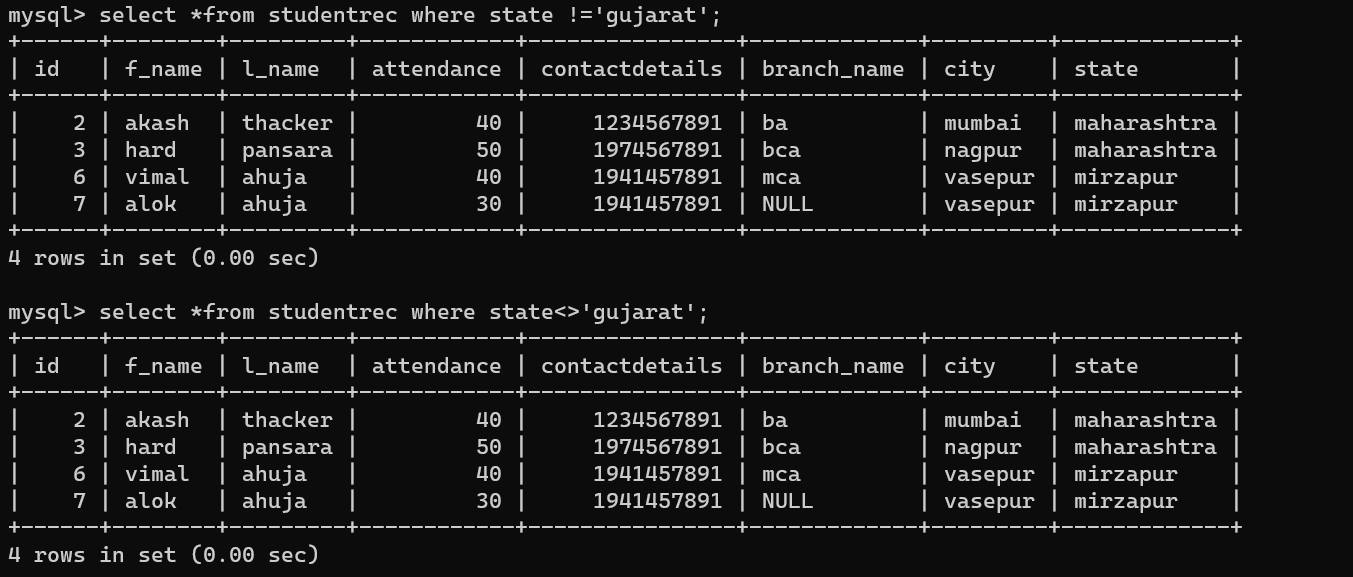
1. Write a query to display student record whose student attendance is 40 or less.

EX :-



1. Write a query to display student record whose student state is not Gujarat.

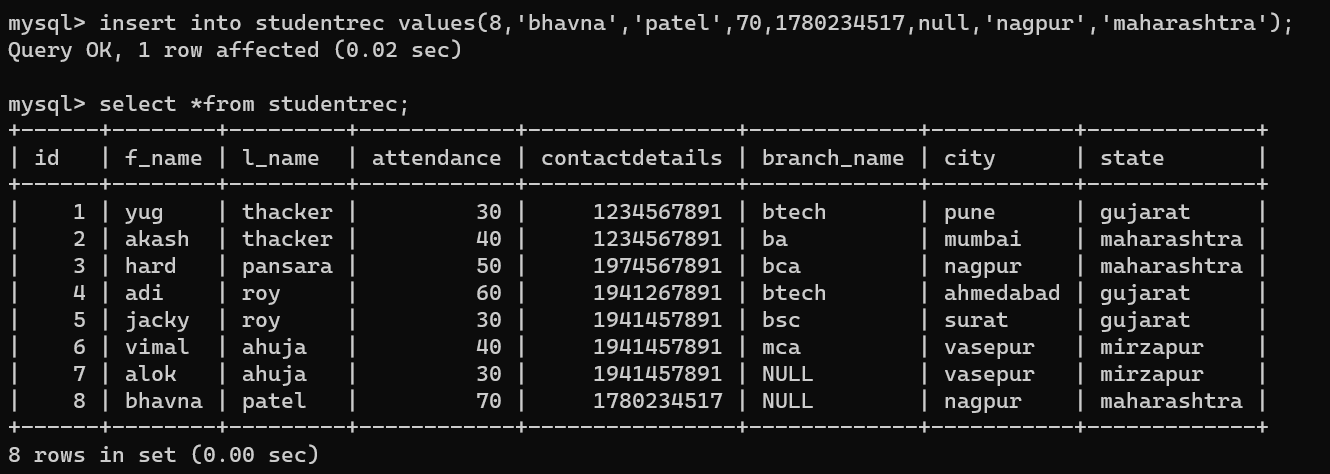
EX :-



**TO add NULL Record**

Syntax :- add null while we insert values without inverted coma

EX :-



**To display NULL Record**

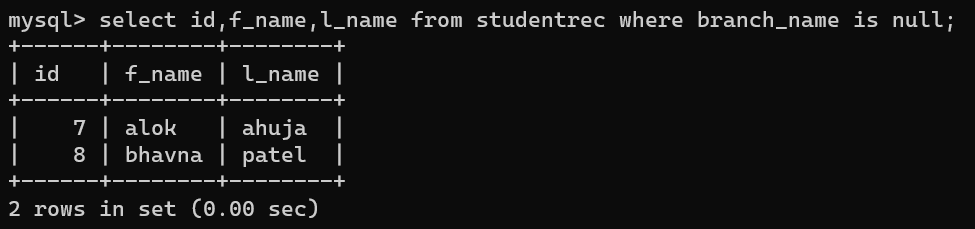
Syntax :- select columnname(we want to show) from tablename where columnname is Null;

OR

select columnname(we want to show) from tablename where columnname is not Null;

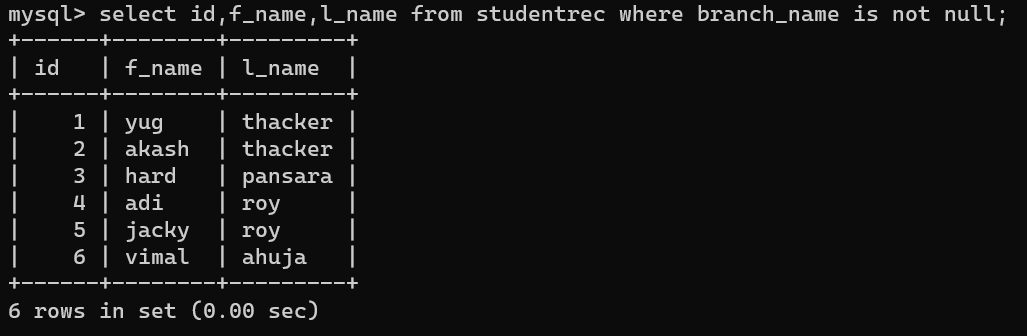
1. Write a query to display record wh0se branch name is null.

EX :-



1. Write a query to display record whose Branch name is not null.

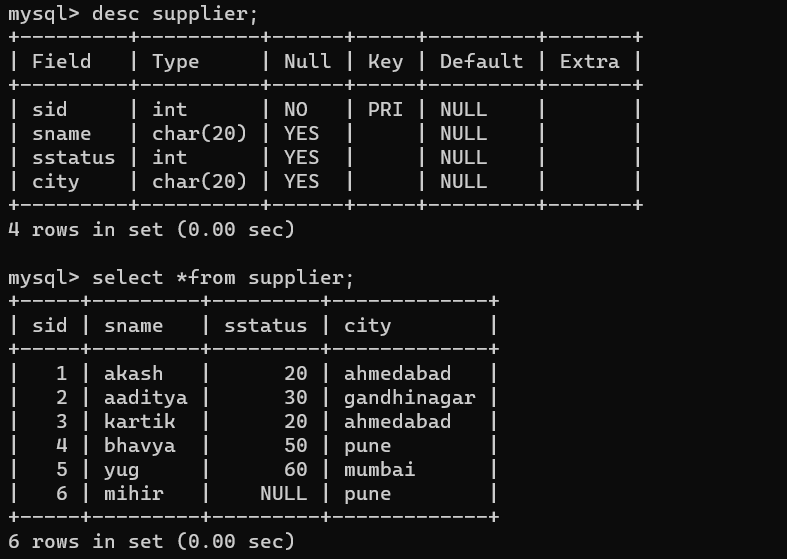
EX :-



**EXAMPLES**

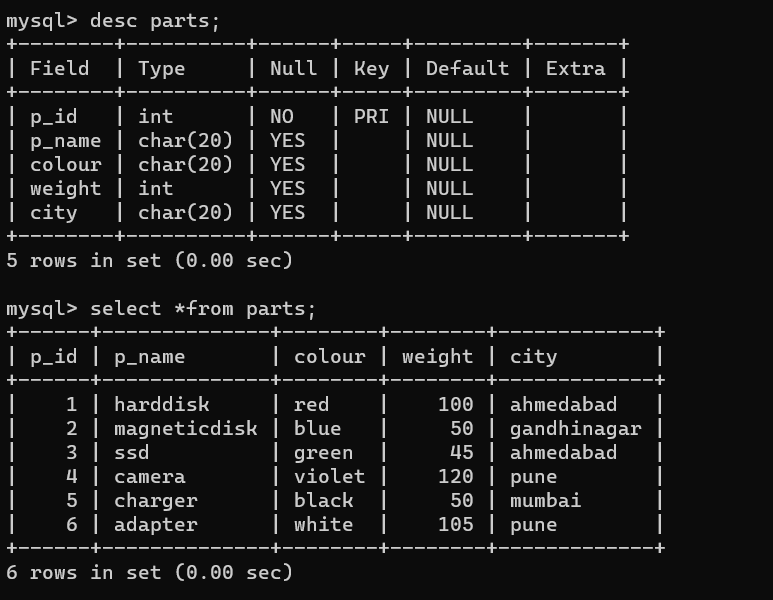
1. Table name supplier (s#,sname ,sstatus,city);

EX :-



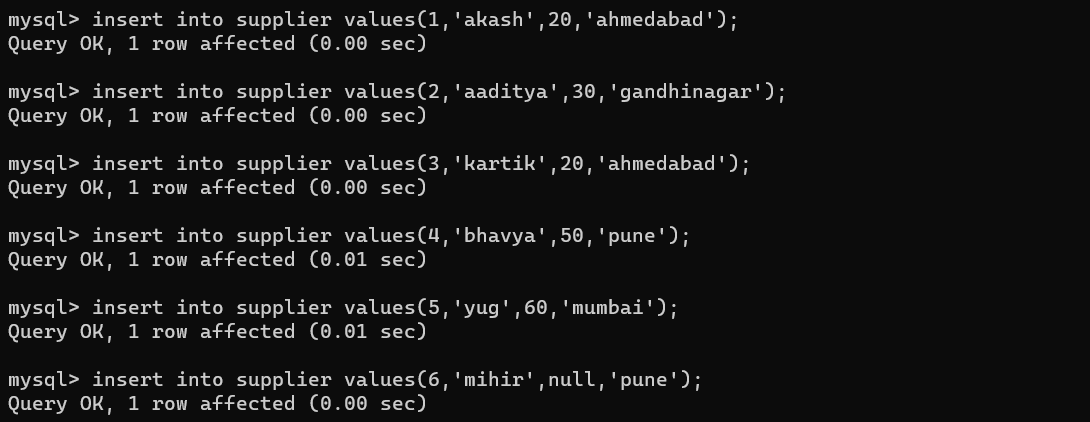
1. Table name parts (partsid#,pname,colour,weight ,city);

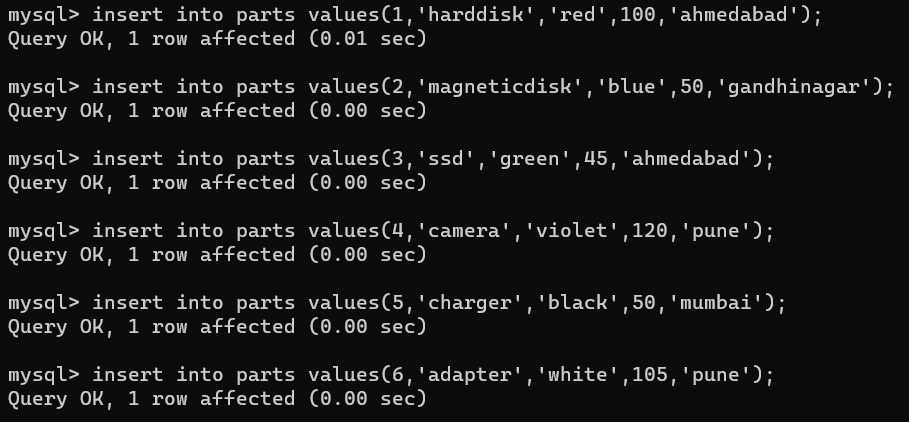
EX :-



1. Write a query to insert records (min 5) to table supplier and parts.

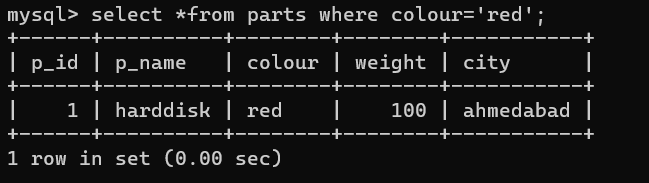
EX :-





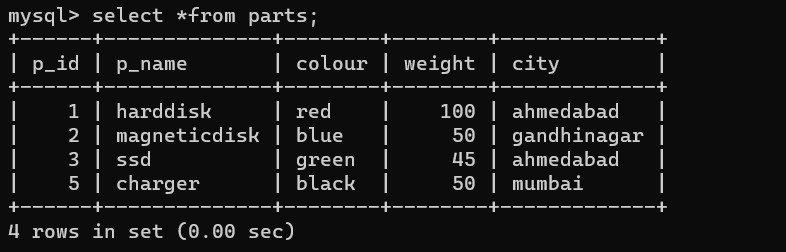
1. Find the name of parts having red colour.

EX :-



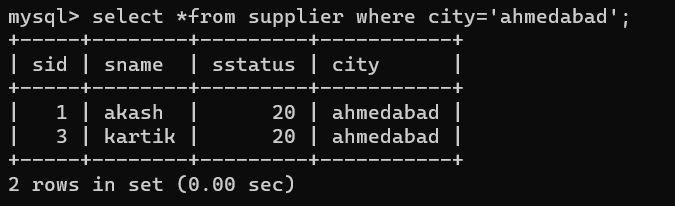
1. delete parts whose weight is more than 100 grams.

EX :-



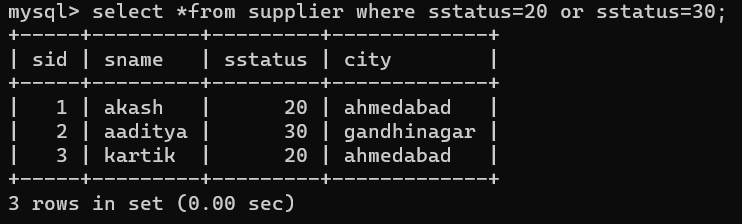
6)display the supplier name which belongs to city ahmedabad.

EX :-



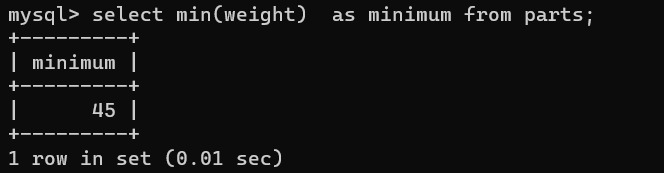
1. Find all suppliers whose status is either 20 or 30.

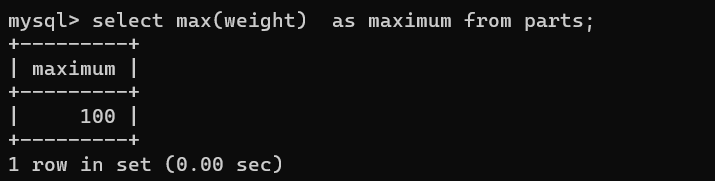
EX :-



1. Find the minimum value and maximum value for parts weight.

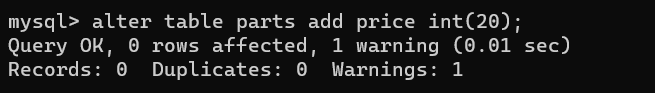
EX :-

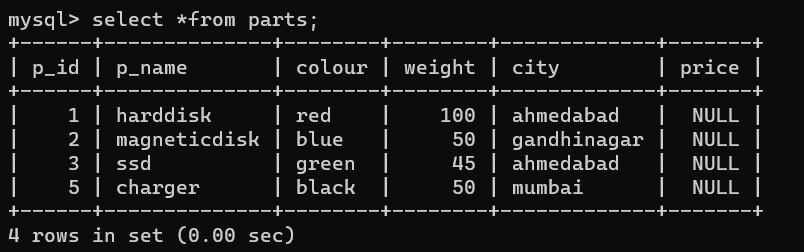




Add new column price to existing table parts.

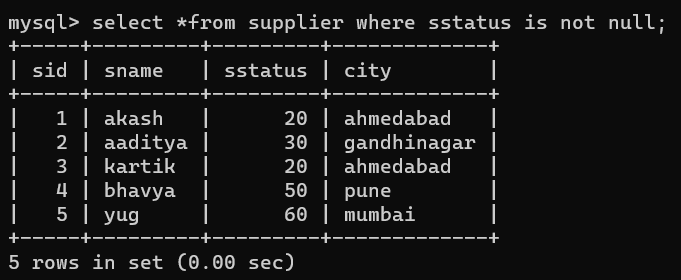
EX :-





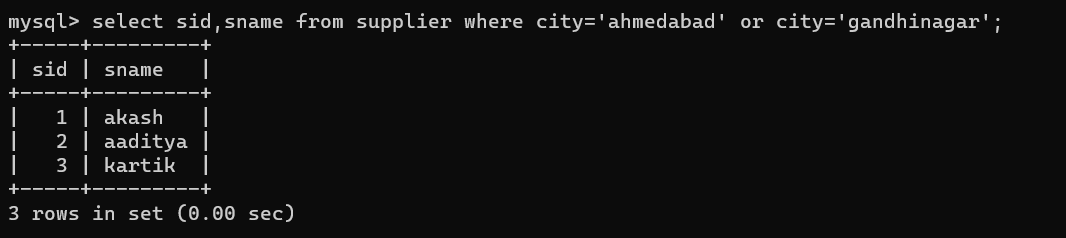
1. Write a sql query to display records of suppliers whose status is not null.

EX :-



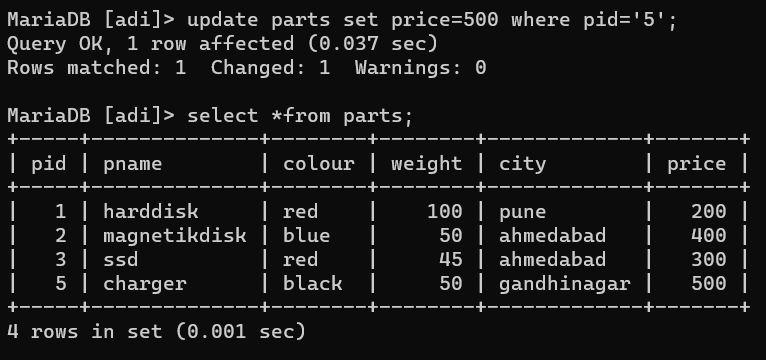
1. Display supplier id and supplier name of supplier who belongs to ahmedabad and gandhinagar city.

EX :-



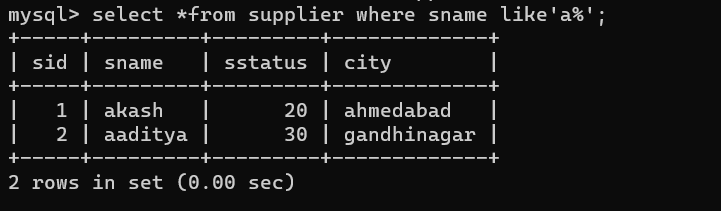
1. Write a query to find unique values for column price.

EX :-



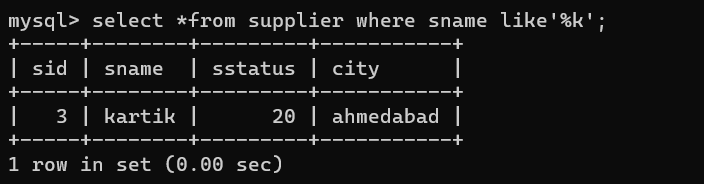
1. Find all the suppliers whose name starts with a.

EX :-



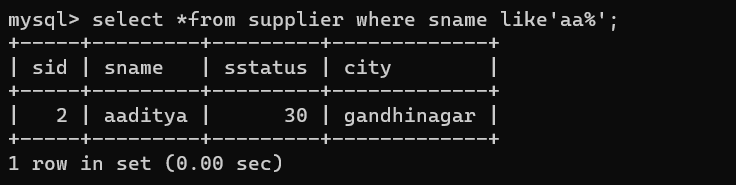
1. Find all the suppliers whose name ends with k.

EX :-



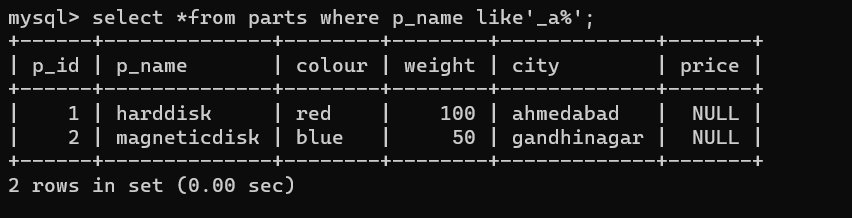
1. Write a query to find all the suppliers whose name starts with aa.

EX :-



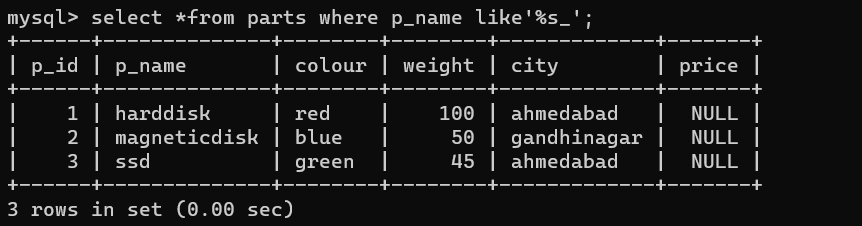
1. Find all the parts whose name's second letter is a.

EX :-



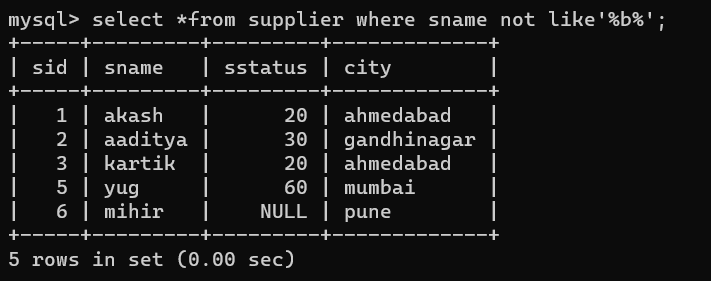
1. Find all the parts whose name's second last letter is s.

EX :-



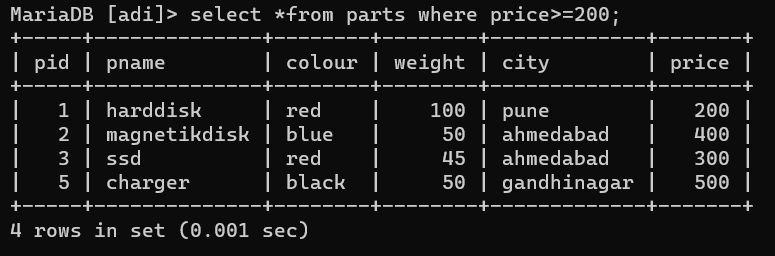
1. Find all the suppliers whose name does not contain b.

EX :-



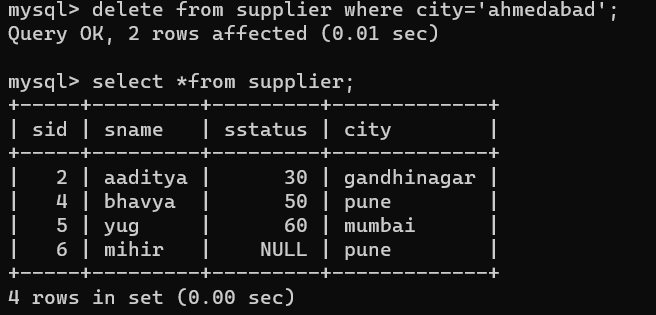
1. Display all the parts whose price is >= 200.

EX :-



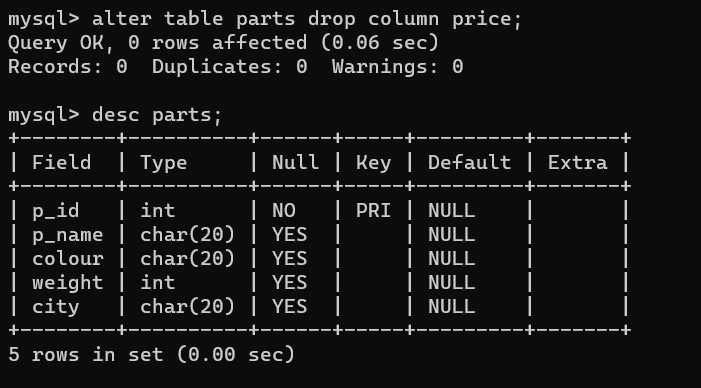
1. Delete all the suppliers who belong to ahmedabad.

EX :-



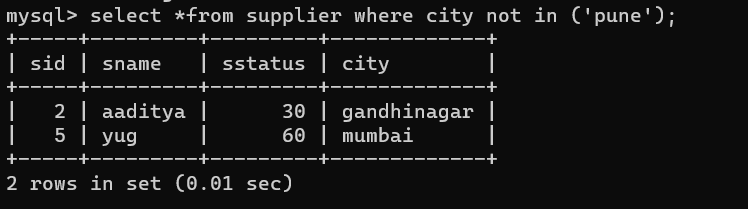
1. Write a query to delete column price from parts table.

EX :-



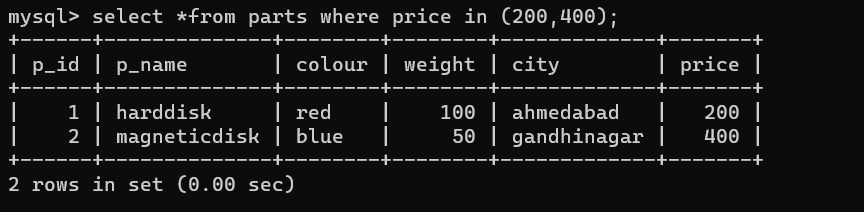
1. Write a query to display supplier records who doesn't belongs to pune. (using not in).

EX :-



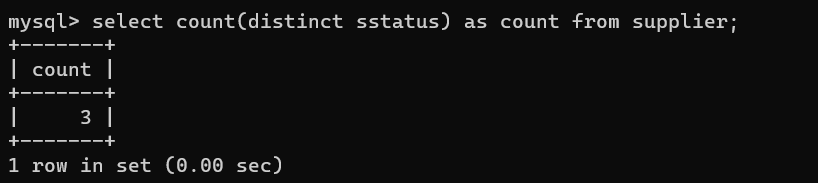
1. Write a query to display records parts name who's price is 100,200,400.

EX :-



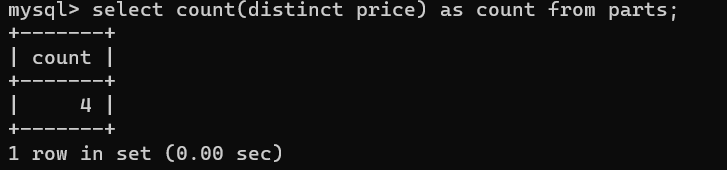
1. Write a query to find unique record count of supplier status.

EX :-



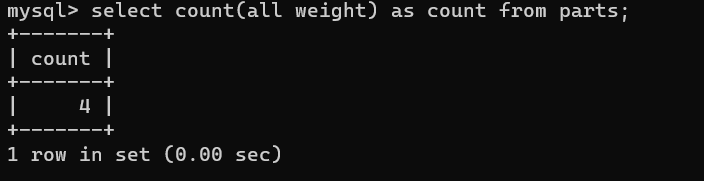
1. Write a query to find total number of unique count for price.

EX :-



1. Write a query to find duplicate records count for field name weight.

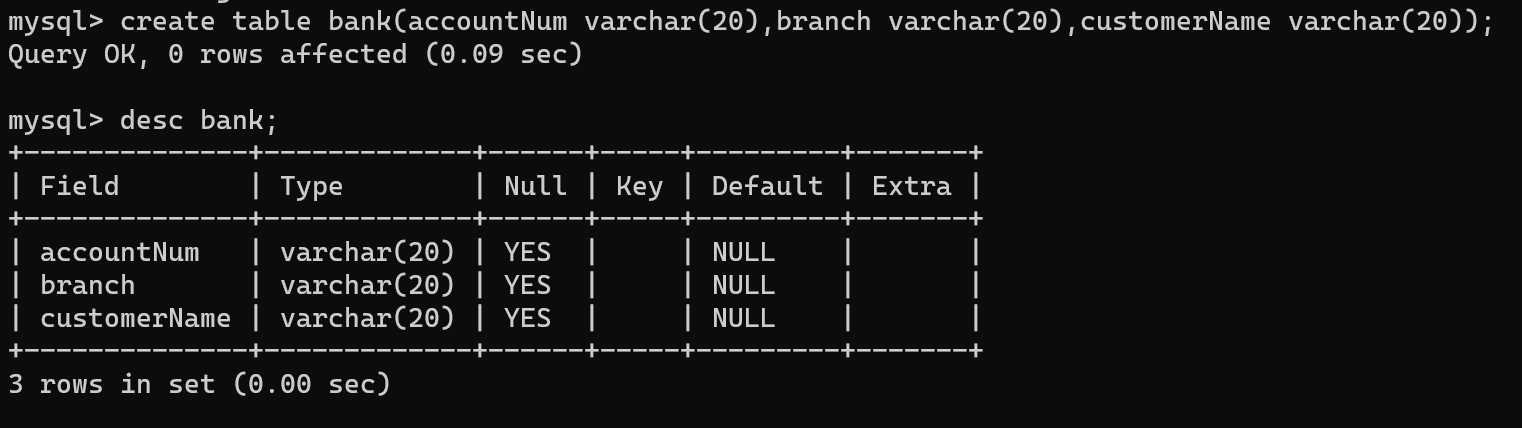
EX :-



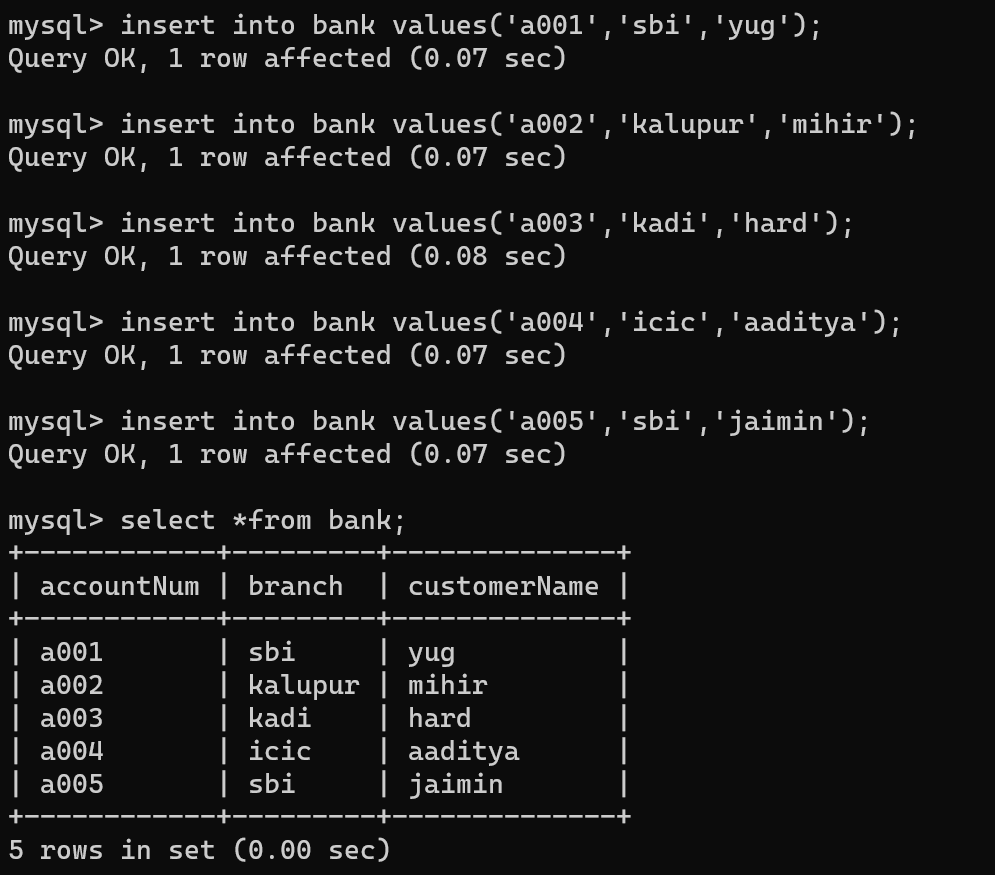
**Update query**

Ques.

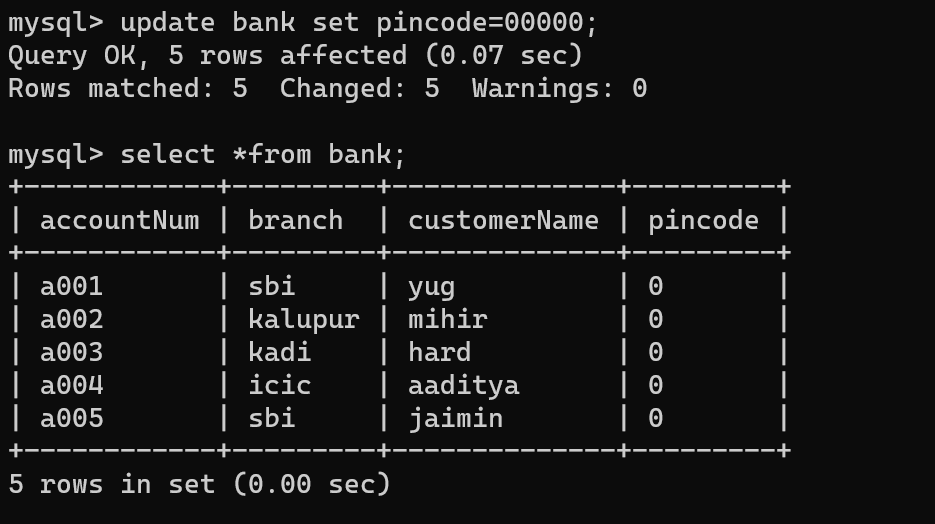
1. Write a query to create a table bank with column name with account no. branch,and customer\_name.



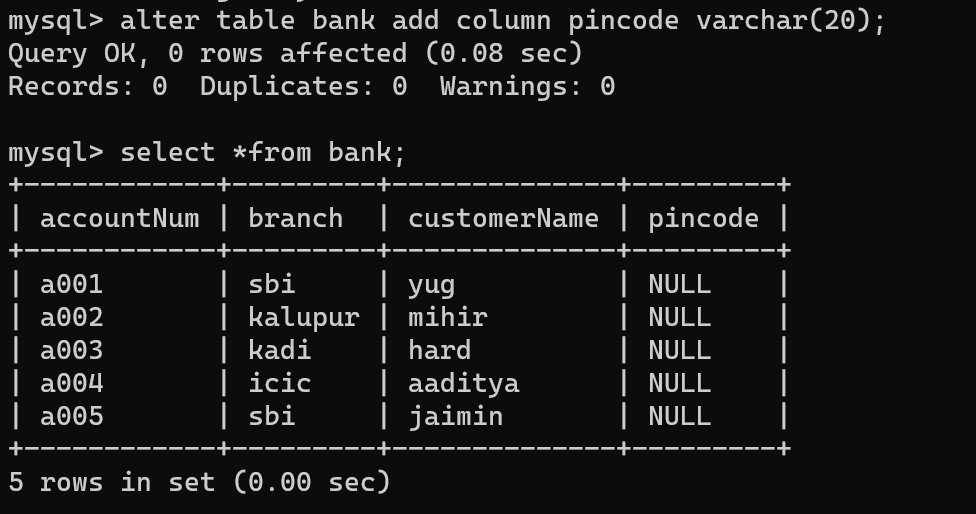
1. Write a query to insert records in bank.



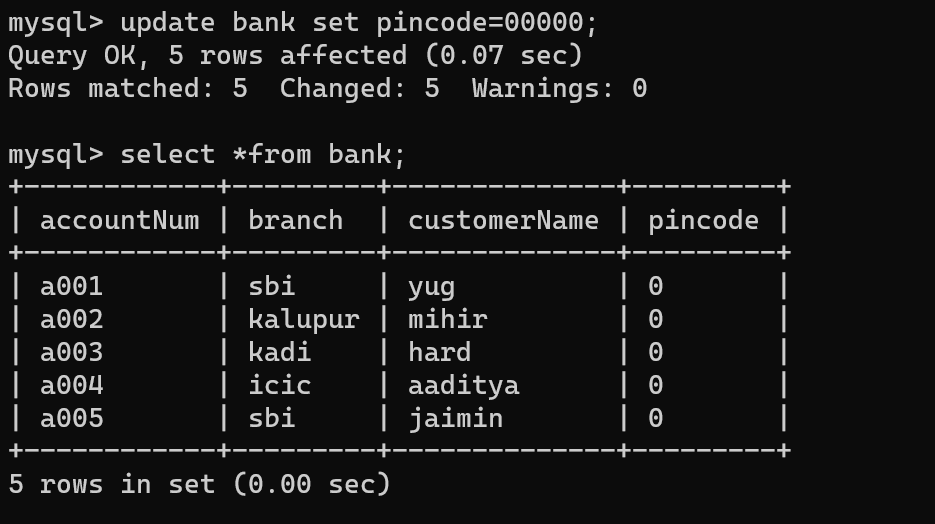
1. Write a query to update the records from the bank table.



1. Add new column pincode to existing table bank.

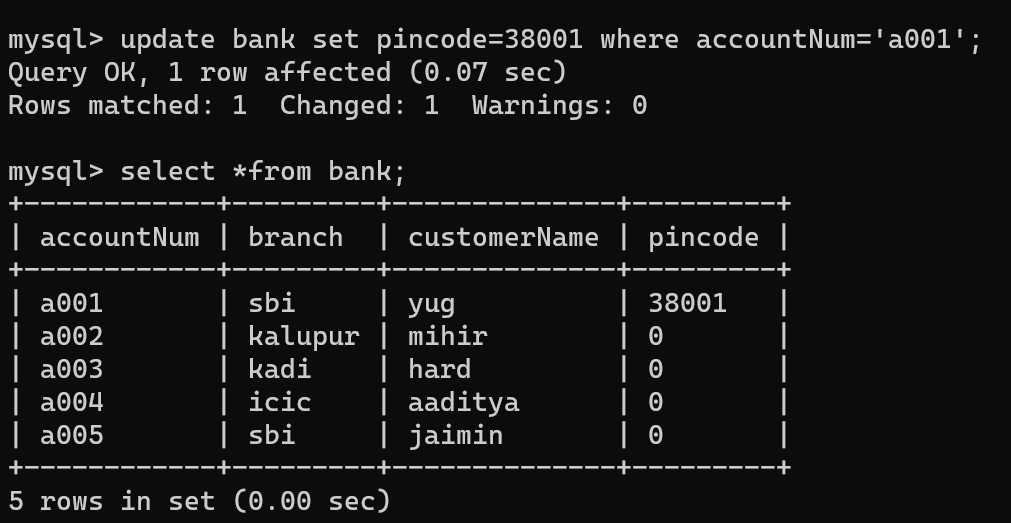


1. Write a query to update the value of pincode.



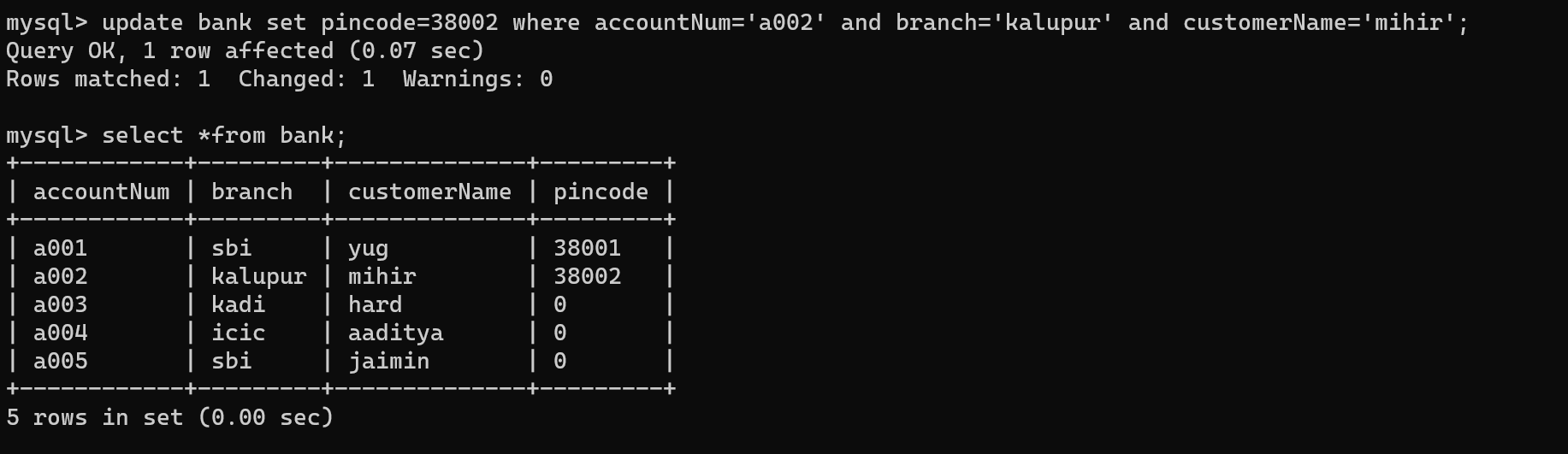
…Update tablename set columname=values;

1. Write a query to update pincode no 380001 whose account number is 1.



…update tablename set pincode=values where columnanme=values;

1. Write a query to update pincode 380002 whose accountno is 2 and customer belongs to kalupur branch and name is mihir.



…update tablename set pincode=380002 where columname=values and columname=values;

**GROUP BY AND ORDER BY**

..Group by -

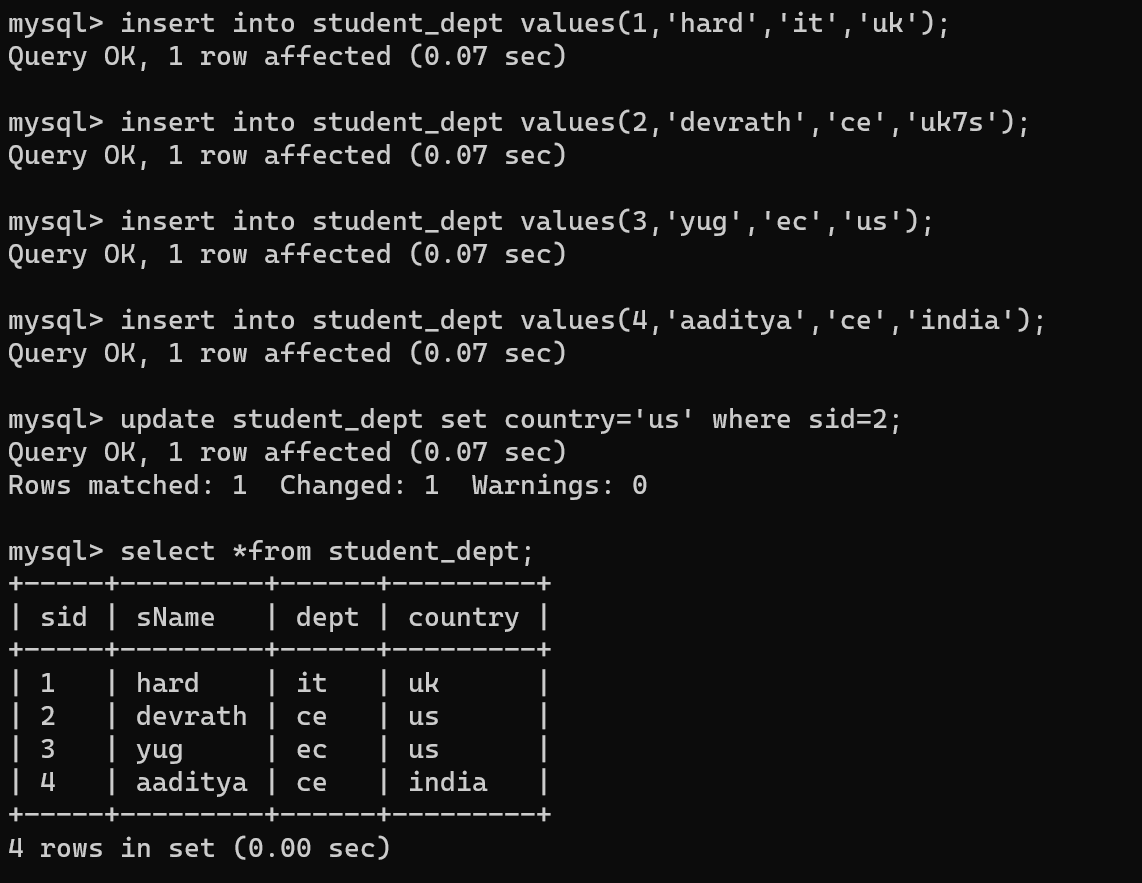
Select columnname from tablename group by columname;

..Order by -

Select columnname from tablename order by columname;

1. Create a table student\_department(sid pk,sname,dept);

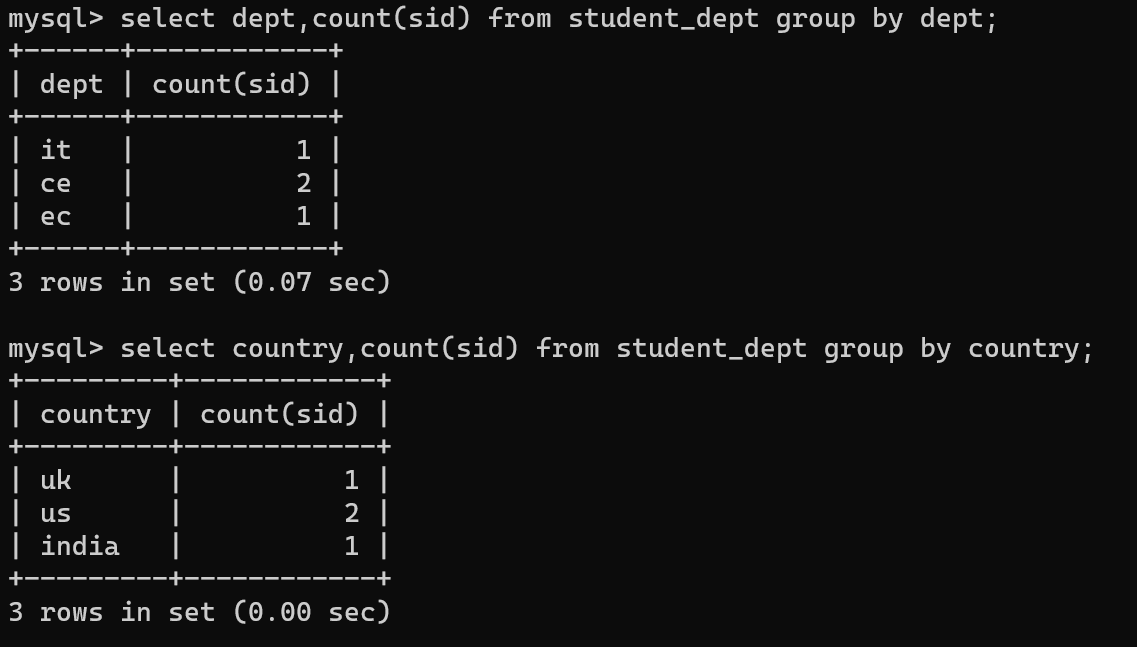




Add column country.

1. Write a query to retrieve no. of students in various department.

…select dept,count(sid) from tablename group by dept;

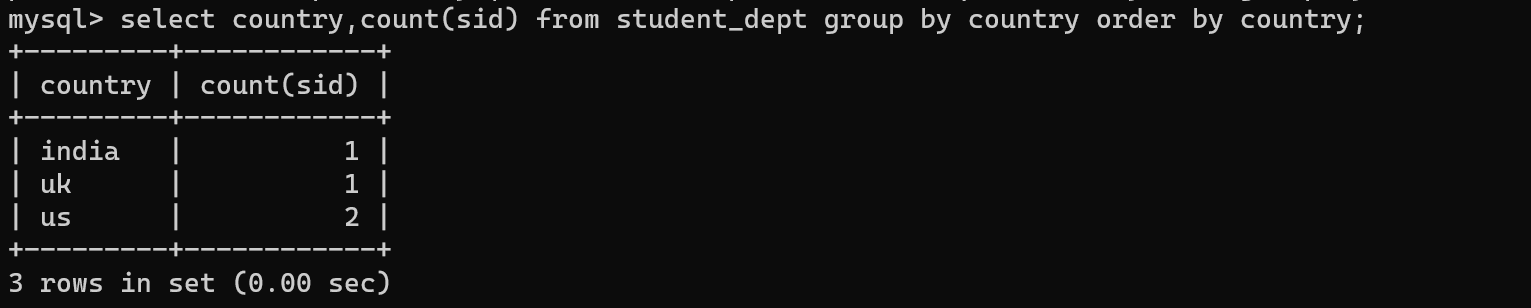


1. Write a query to retrieve no. of students in various country.

…select country,count(sid) from student\_department group by country;

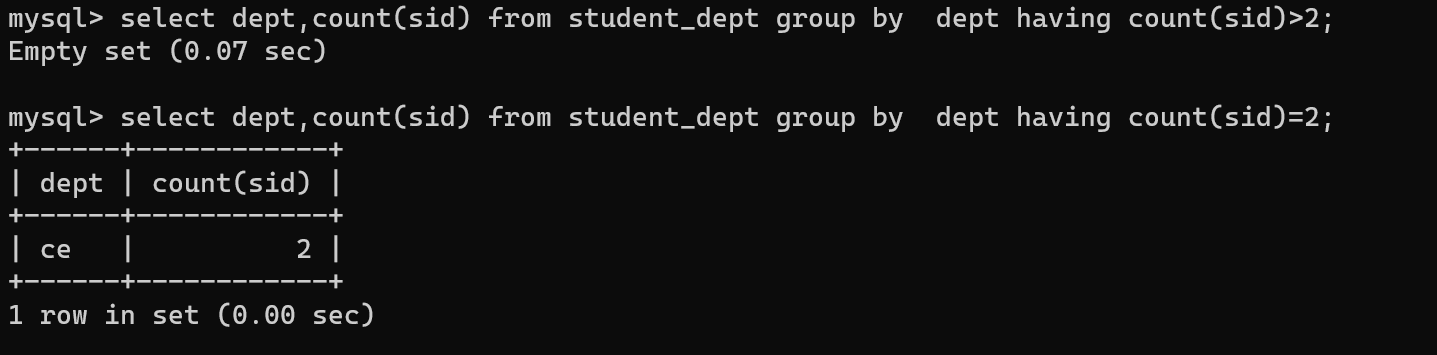
Order by

Group by and order by use



**HAVING**

1. Write a query to retrive department having more or equal to 2 students in it.



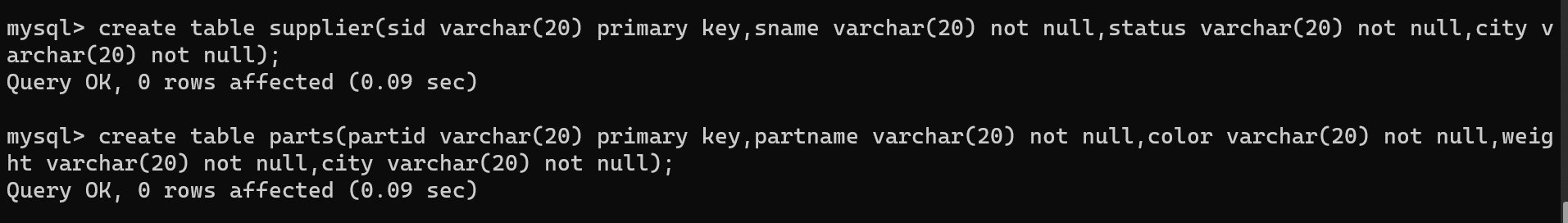
**PRACTICE QUESTION**

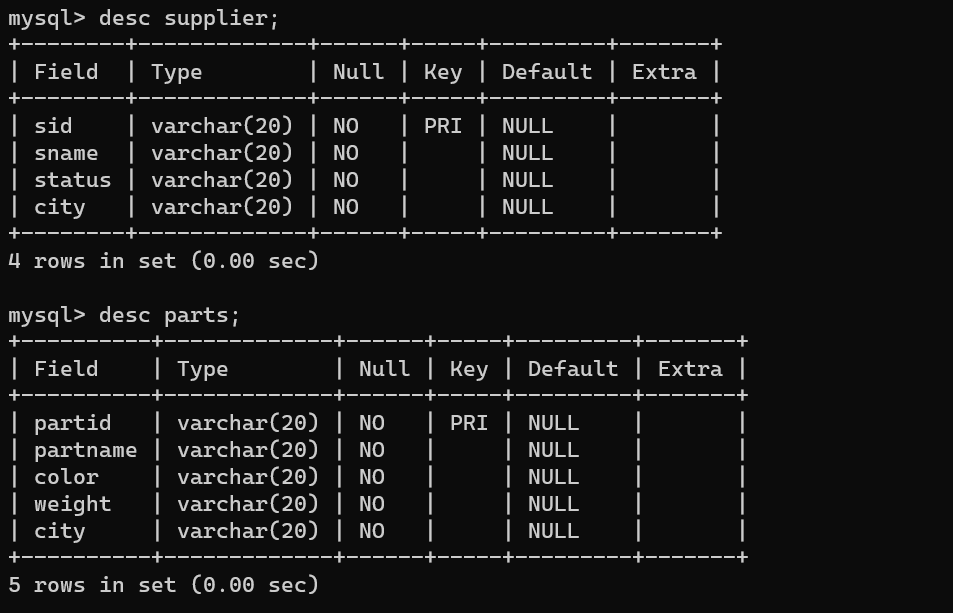
Create the table.

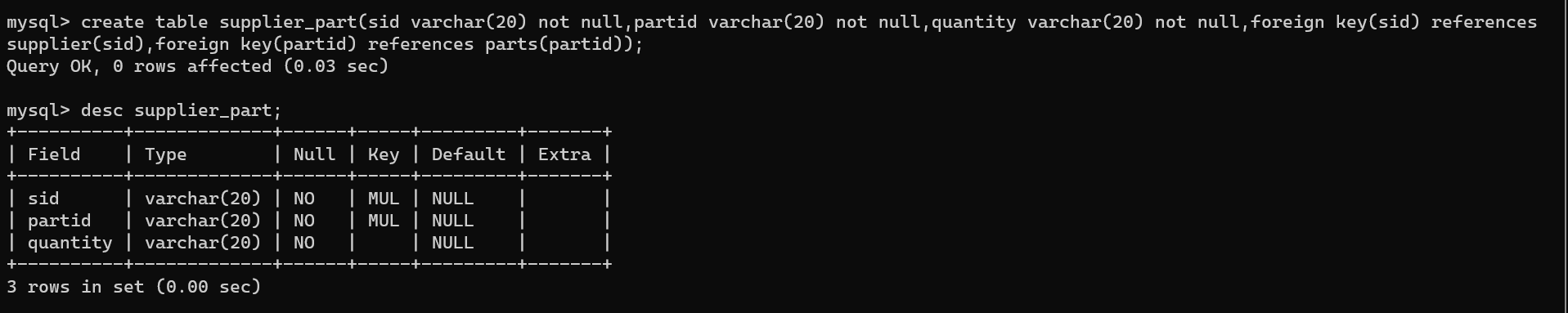
Supplier – sid , sname , status , city

Parts – partid , partname , color , weight , city

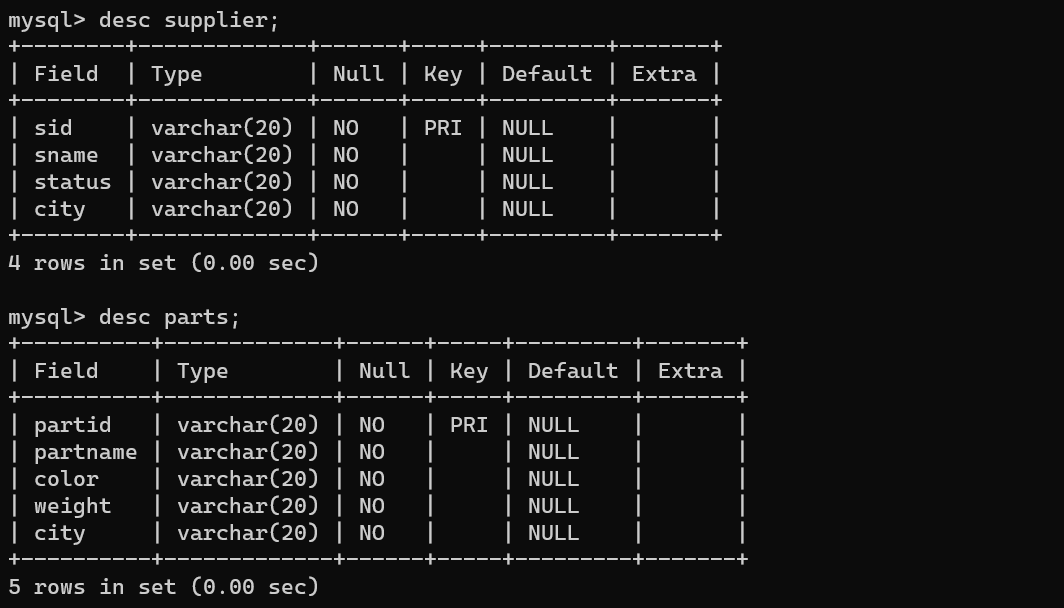
Supplier\_parts – sid , partid , quantity

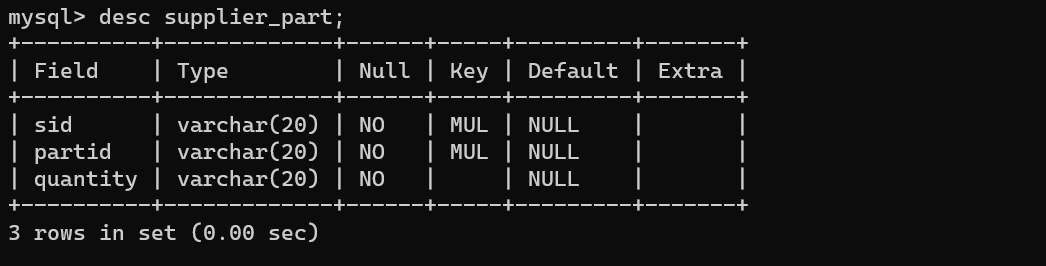




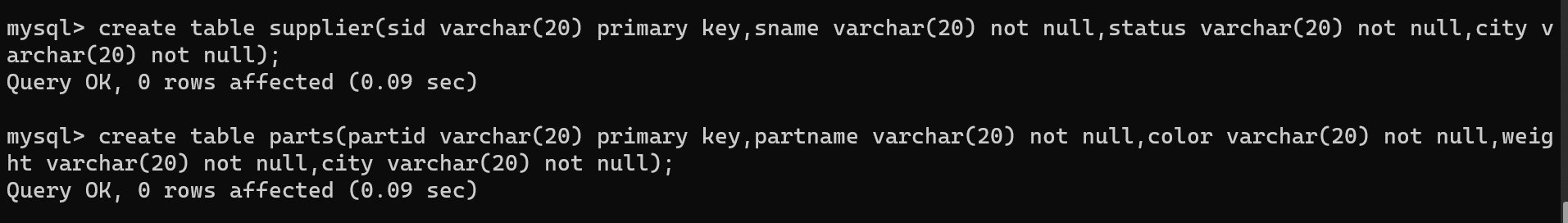


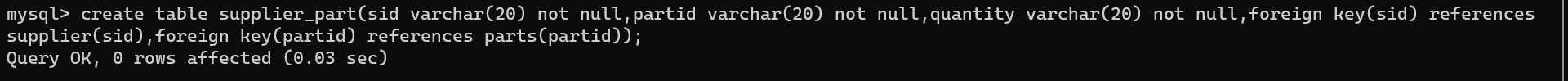
1. Identify primary and foreign key from above mention table.



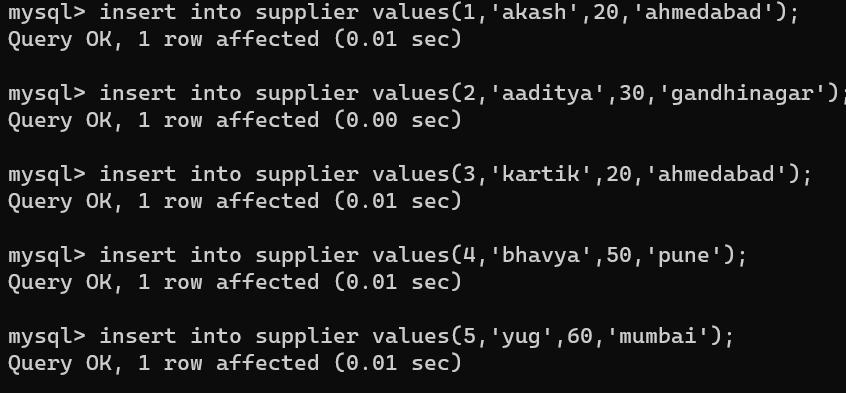


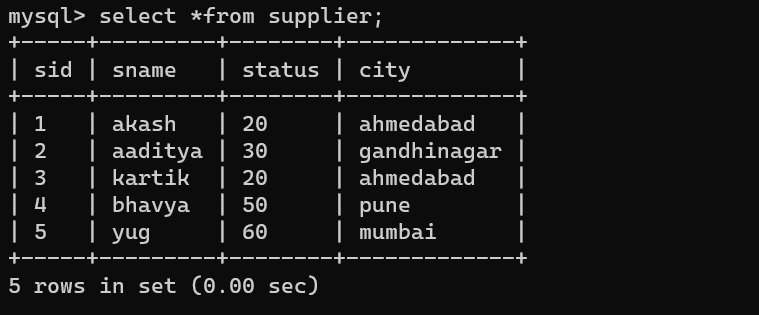
1. WRITE A QUERY TO CREATE TABLES WITH PRIMARY KEY, SECONDARY KEY, NOT NULL

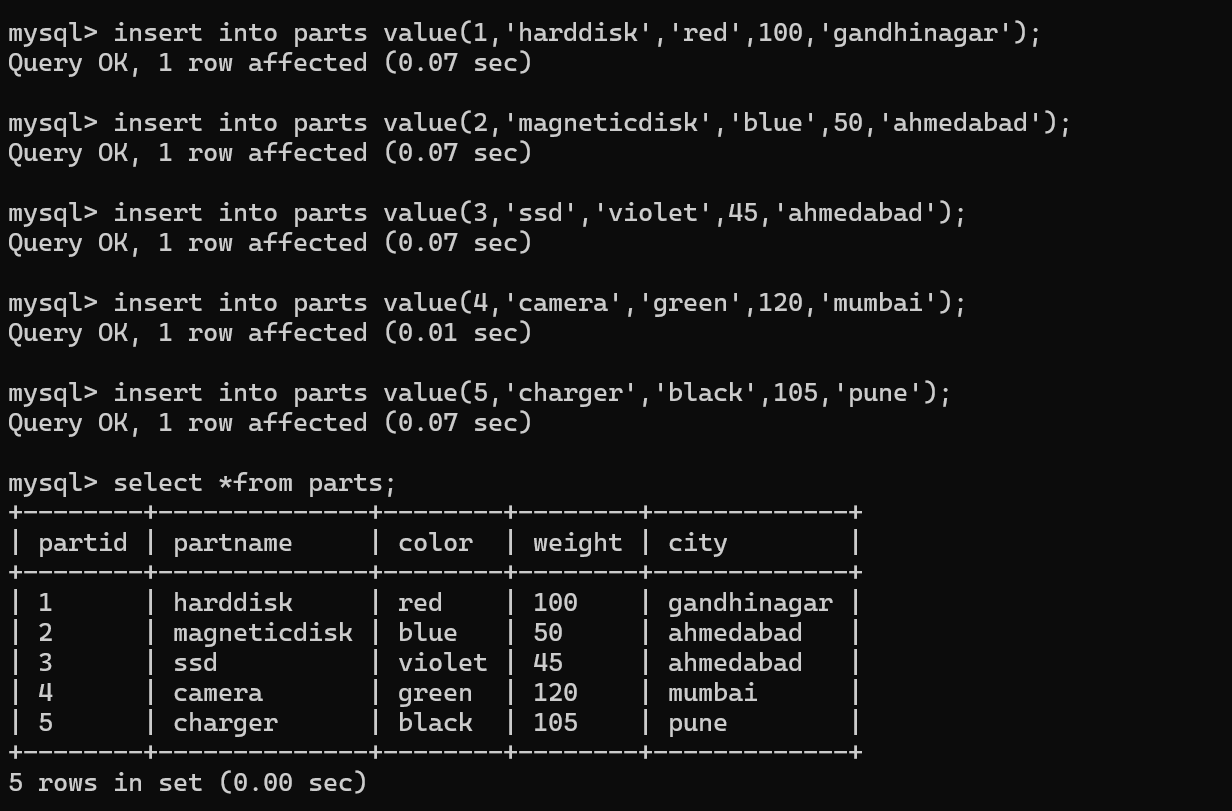


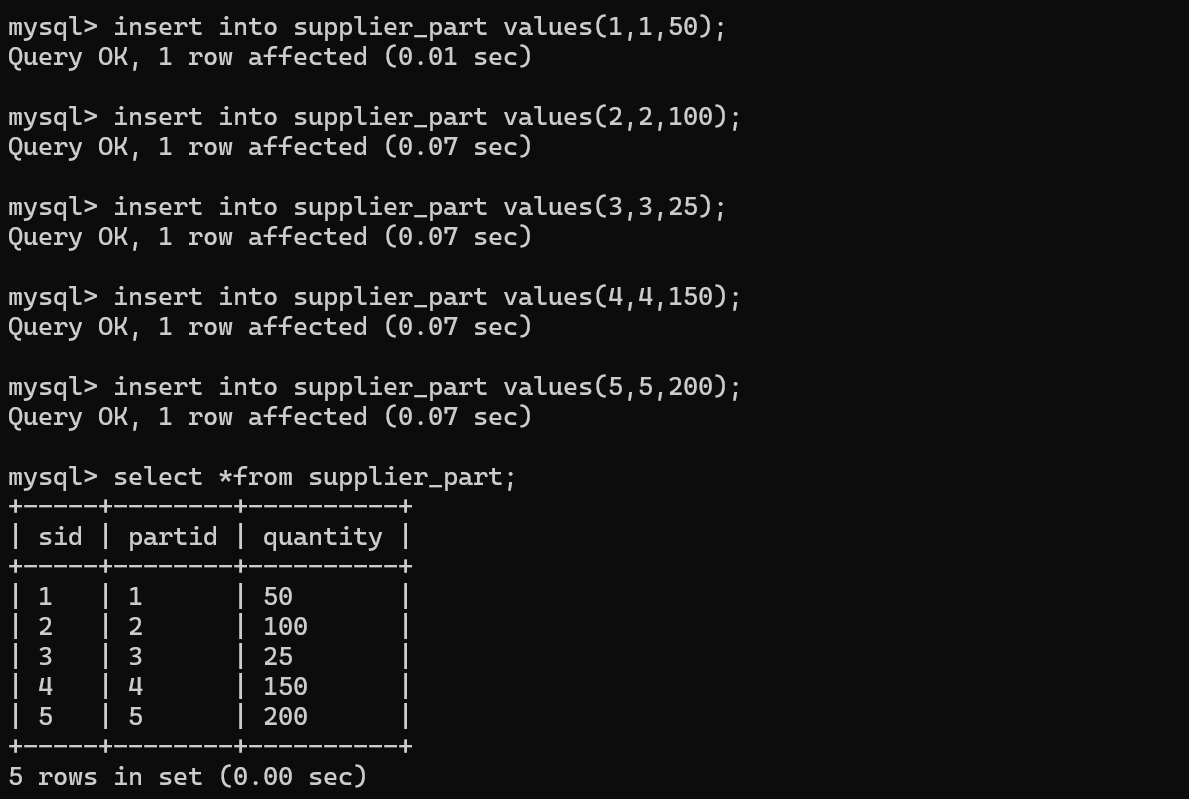


1. INSERT RECORDS TO TABLES

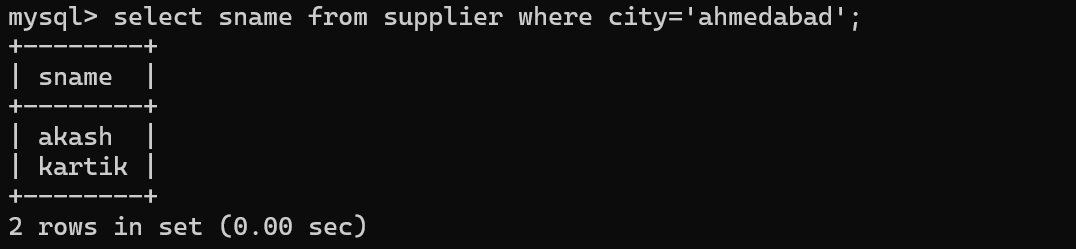




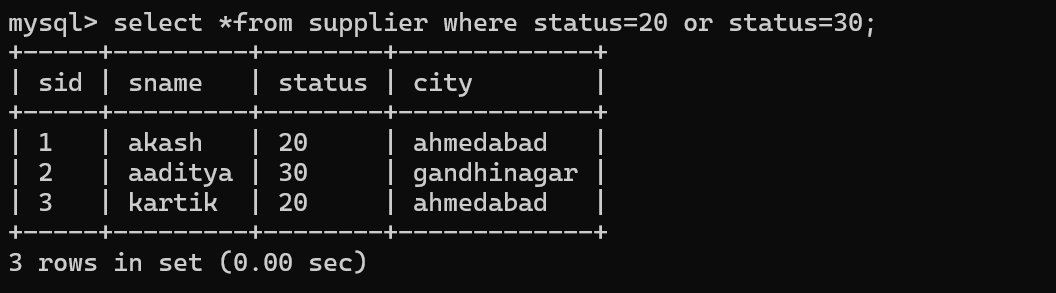




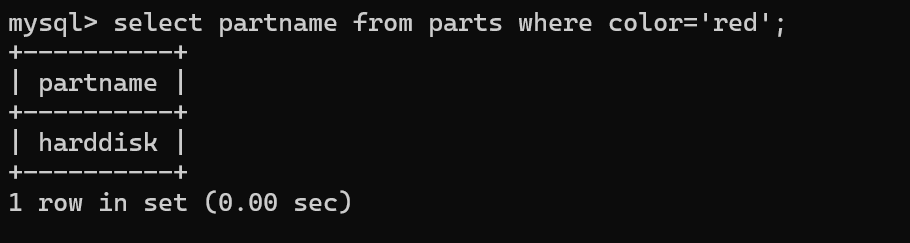
1. DISPLAY THE NAME OF SUPPLIER WHO LIVES IN AHMEDABAD.



1. FIND ALL SUPPLIERS WHOSE STATUS IS EITHER 20 OR 30.



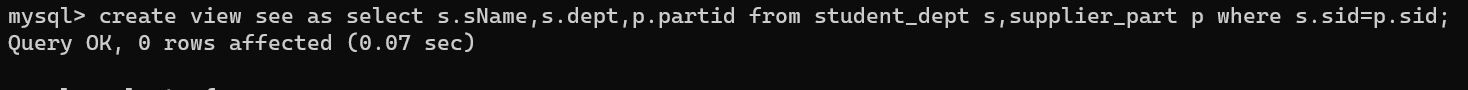
1. FIND THE NAME OF PARTS HAVING RED COLORS.



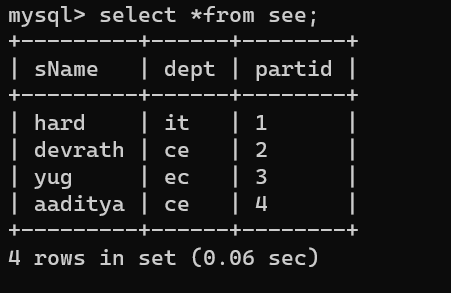
**VIEW**

VIEW : VIEW is a virtual table based on results set of An sql statement.

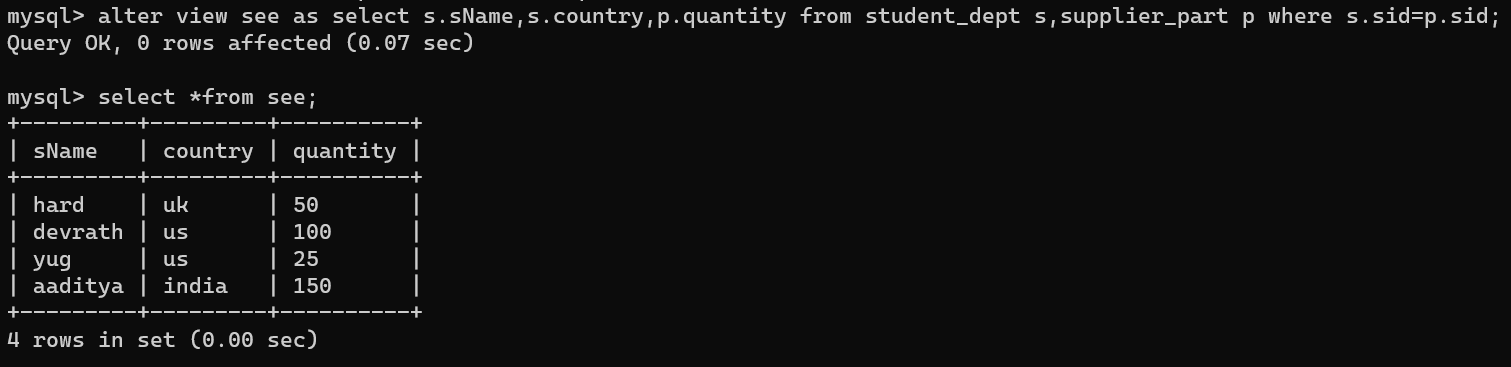
SYNTAX : CREATE OR (REPLACE) VIEW VIEWNAME AS SELECT COLUMN FROM TABLES [WHERE CONDITION]



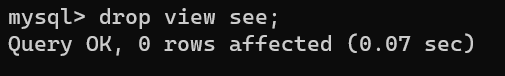
SELECT \* FROM VIEW NAME



ALTER VIEW VIEWNAME AS SELECT COLUMNS FROME TABLE WHERE CONDITION



DROP VIEW VIEWNAME



EX :- CREATE VIEW TRAINER AS SELECT c.course\_name,c.trainer,t.email from course c,contact t where c.id=t.id;

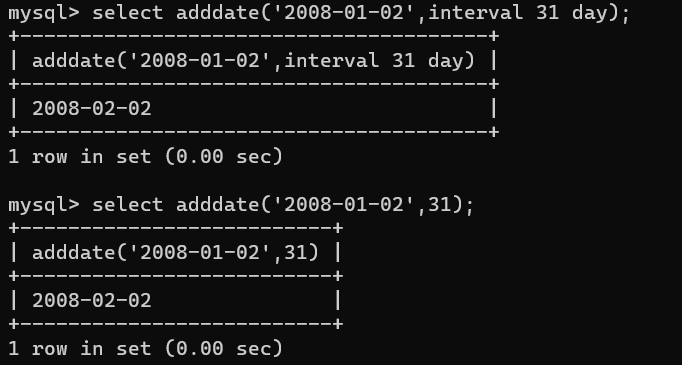
**Time and Date Functions**

1. Adddate()

* Add time values (intervals) to a date value

Syntax :- [**ADDDATE(*date*,INTERVAL *expr* *unit*)**](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_adddate)**,**[**ADDDATE(*expr*,*days*)**](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_adddate)

EX :-



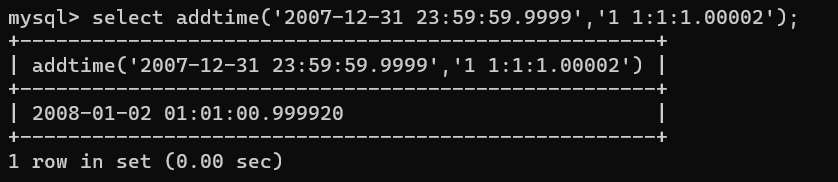
1. Addtime()

* Add time

Syntax :- [ADDTIME()](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_addtime) adds ***expr2*** to ***expr1*** and returns the result. ***expr1*** is a time or datetime expression, and ***expr2*** is a time expression.

EX :-

00

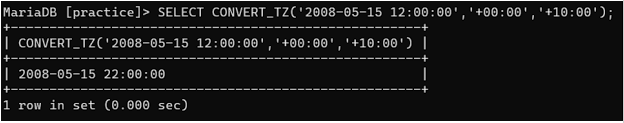


1. Convert\_tz()

* Convert from one time zone to another

Syntax :- [CONVERT\_TZ(***dt***,***from\_tz***,***to\_tz***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_convert-tz)

EX :-

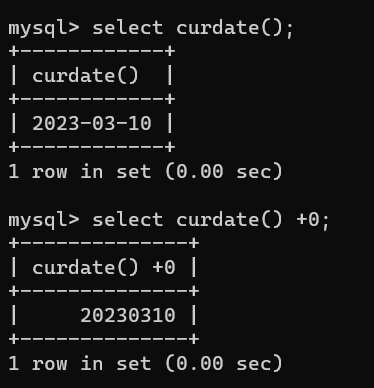


1. Curdate()

* Return the current date

Syntax :- Returns the current date as a value in '***YYYY-MM-DD***' or ***YYYYMMDD*** format, depending on whether the function is used in string or numeric context.

EX :-

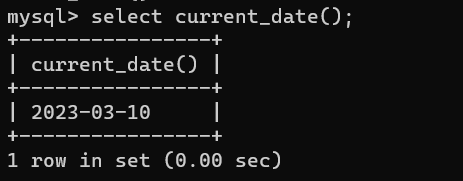


1. Current\_date()

* Return the current date

Syntax :- [CURRENT\_DATE](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_current-date), [CURRENT\_DATE()](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_current-date)

EX :-

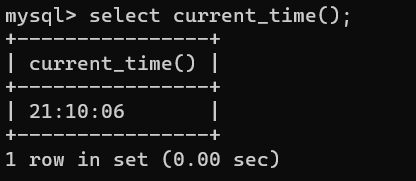


1. Current\_time()

* Return the current time

Syntax :- [CURRENT\_TIME](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_current-time), [CURRENT\_TIME()](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_current-time)

EX :-

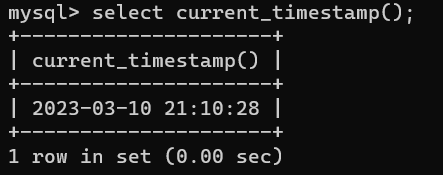


1. Current\_timestamp()

* Return the current date and time

Syntax :- [CURRENT\_TIMESTAMP](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_current-timestamp), [CURRENT\_TIMESTAMP()](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_current-timestamp)

EX :-

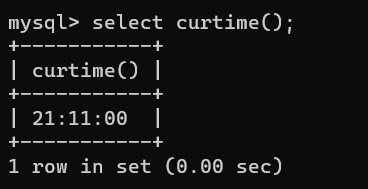


1. Curtime()

* Return the current time

Syntax :- [CURTIME()](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_curtime)

EX :-

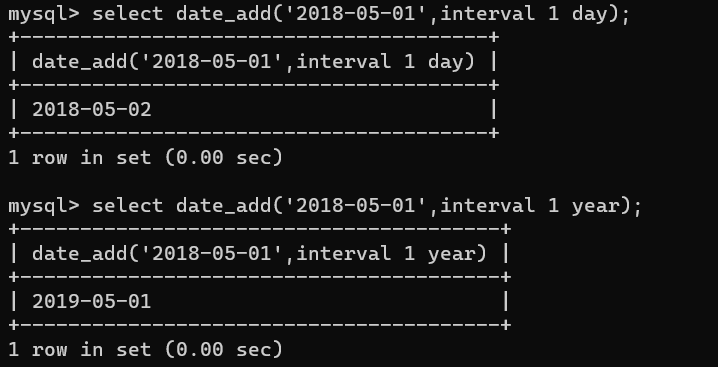


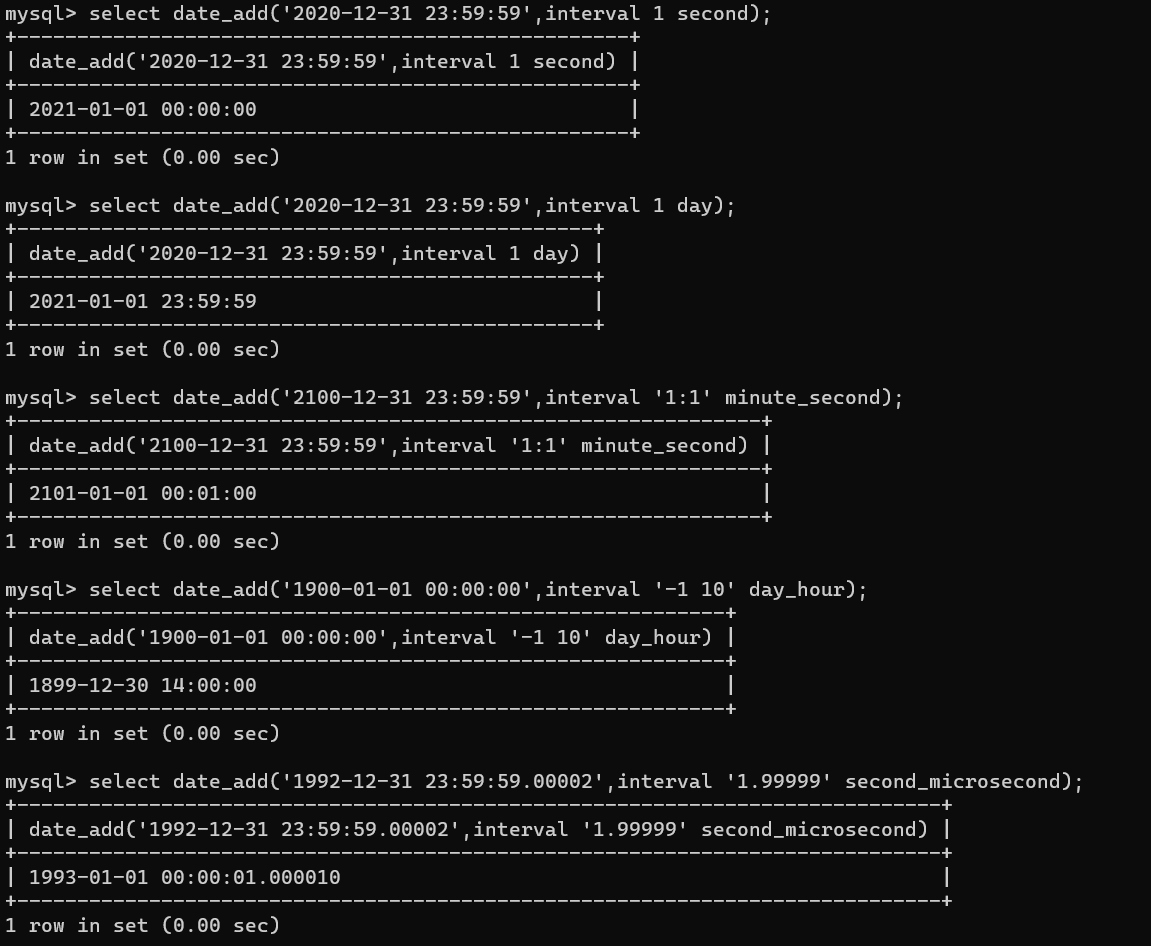
1. Date\_add()

* Add time values (intervals) to a date value

Syntax :- [DATE\_ADD(***date***,INTERVAL ***expr*** ***unit***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_date-add)

EX :-





1. Date\_format()

* Format date as specified

Syntax :- [DATE\_FORMAT(***date***,***format***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_date-format)

| **Specifier** | **Description** |
| --- | --- |
| %a | Abbreviated weekday name (Sun..Sat) |
| %b | Abbreviated month name (Jan..Dec) |
| %c | Month, numeric (0..12) |
| %D | Day of the month with English suffix (0th, 1st, 2nd, 3rd, …) |
| %d | Day of the month, numeric (00..31) |
| %e | Day of the month, numeric (0..31) |
| %f | Microseconds (000000..999999) |
| %H | Hour (00..23) |
| %h | Hour (01..12) |
| %I | Hour (01..12) |
| %i | Minutes, numeric (00..59) |
| %j | Day of year (001..366) |
| %k | Hour (0..23) |
| %l | Hour (1..12) |
| %M | Month name (January..December) |
| %m | Month, numeric (00..12) |
| %p | AM or PM |
| %r | Time, 12-hour (***hh:mm:ss*** followed by AM or PM) |
| %S | Seconds (00..59) |
| %s | Seconds (00..59) |
| %T | Time, 24-hour (***hh:mm:ss***) |
| %U | Week (00..53), where Sunday is the first day of the week; [WEEK()](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_week) mode 0 |
| %u | Week (00..53), where Monday is the first day of the week; [WEEK()](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_week) mode 1 |
| %V | Week (01..53), where Sunday is the first day of the week; [WEEK()](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_week) mode 2; used with %X |
| %v | Week (01..53), where Monday is the first day of the week; [WEEK()](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_week) mode 3; used with %x |
| %W | Weekday name (Sunday..Saturday) |
| %w | Day of the week (0=Sunday..6=Saturday) |
| %X | Year for the week where Sunday is the first day of the week, numeric, four digits; used with %V |
| %x | Year for the week, where Monday is the first day of the week, numeric, four digits; used with %v |
| %Y | Year, numeric, four digits |
| %y | Year, numeric (two digits) |
| %% | A literal % character |
| %***x*** | ***x***, for any “***x***” not listed above |

EX :-

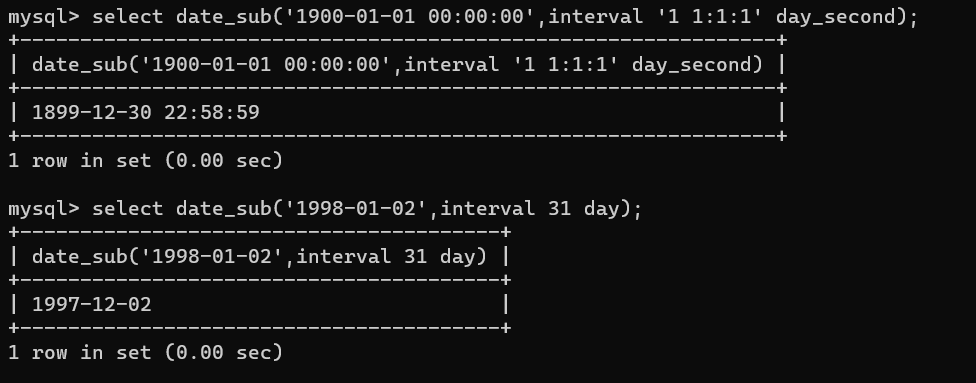


1. Date\_sub()

* Subtract a time value (interval) from a date

Syntax :-  [DATE\_SUB(***date***,INTERVAL ***expr*** ***unit***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_date-sub)

EX :-

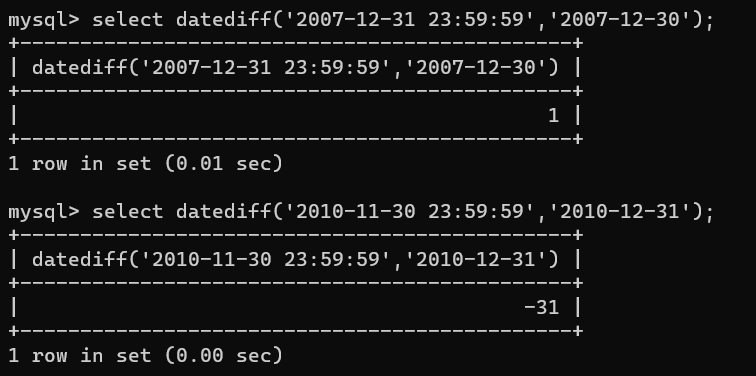


1. Datediff()

* Subtract two dates

Syntax :- [DATEDIFF(***expr1***,***expr2***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_datediff)

EX :-

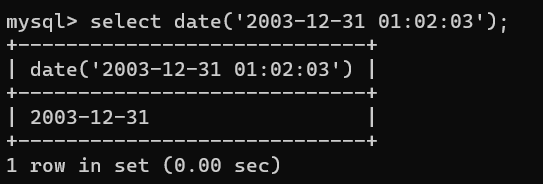


1. Date()

* Extract the date part of a date or datetime expression

Syntax :- [DATE(***expr***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_date)

EX :-

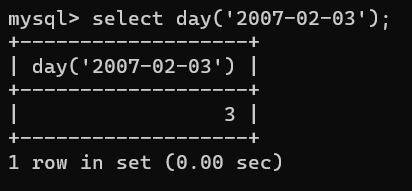


1. Day()

* Return the day of the month (0-31)

Syntax :- [DAY(***date***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_day)

EX :-

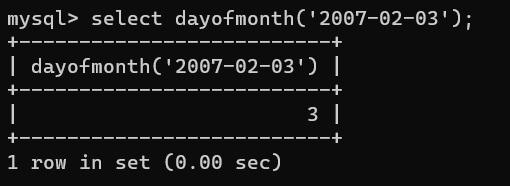


1. Dayofmonth()

* Return the day of the month (0-31)

Syntax :- [DAYOFMONTH(***date***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_dayofmonth)

EX :-

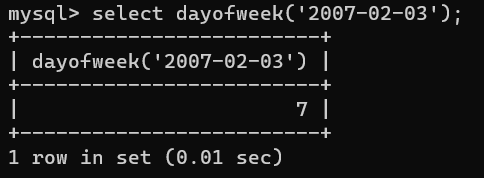


1. Dayofweek()

* Return the weekday index of the argument

Syntax :- [DAYOFWEEK(***date***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_dayofweek)

EX :-

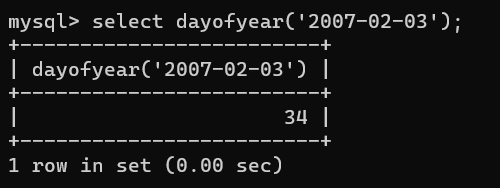


1. Dayofyear()

* Return the day of the year (1-366)

Syntax :- [DAYOFYEAR(***date***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_dayofyear)

EX :-

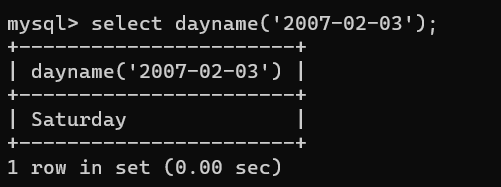


1. Dayname()

* Return the name of the weekday

Syntax :- [DAYNAME(***date***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_dayname)

EX :-

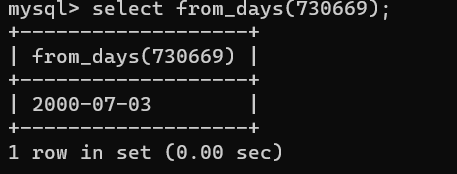


1. From\_days()

* Convert a day number to a date

Syntax :- [FROM\_DAYS(***N***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_from-days)

EX :-

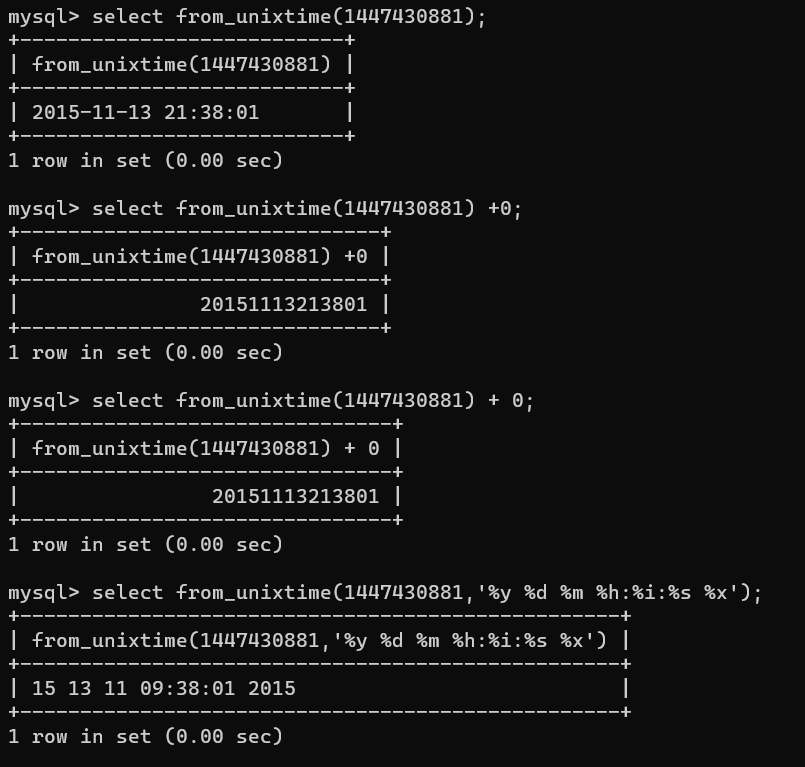


1. From\_unixtime()

* Format Unix timestamp as a date

Syntax :- [FROM\_UNIXTIME(***unix\_timestamp***[,***format***])](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_from-unixtime)

EX :-

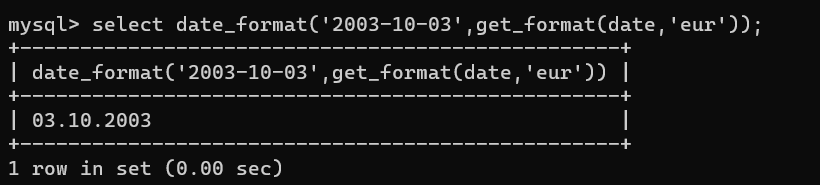


1. Get\_format()

* Return a date format string

Syntax :- [GET\_FORMAT({DATE|TIME|DATETIME}, {'EUR'|'USA'|'JIS'|'ISO'|'INTERNAL'})](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_get-format)

EX :-

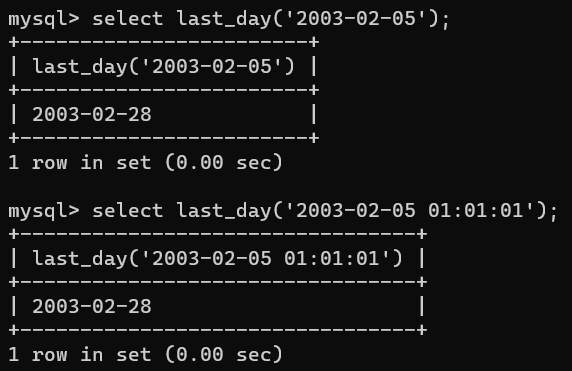


1. Last\_day()

* Return the last day of the month for the argument

Syntax :- [LAST\_DAY(***date***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_last-day)

EX :-

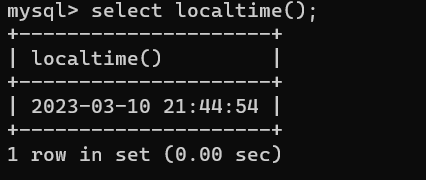


1. Localtime()

* Return the current date and time

Syntax :- [LOCALTIME](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_localtime), [LOCALTIME([***fsp***])](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_localtime)

EX :-

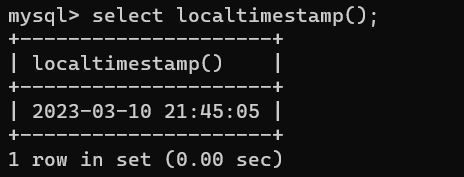


1. Localtimestamp()

* Return the current date and time

Syntax :- [LOCALTIMESTAMP](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_localtimestamp), [LOCALTIMESTAMP()](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_localtimestamp)

EX :-

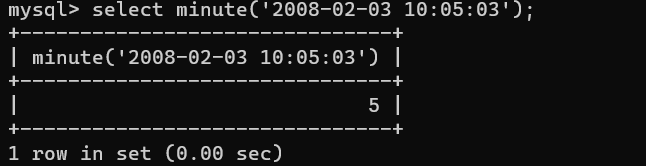


1. Minute()

* Return the minute from the argument

Syntax :- [MINUTE(***time***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_minute)

EX :-

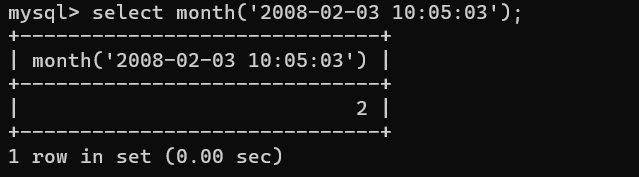


1. Month()

* Return the month from the date passed

Syntax :- [MONTH(***date***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_month)

EX :-

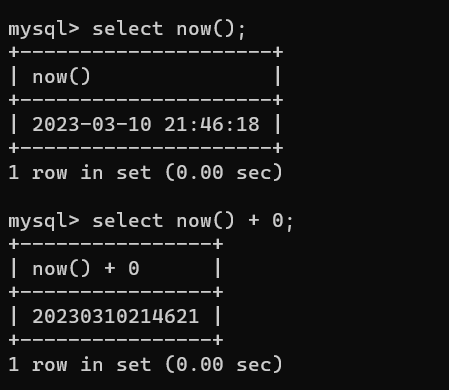


1. Now()

* Return the current date and time

Syntax :- [NOW()](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_now)

EX :-

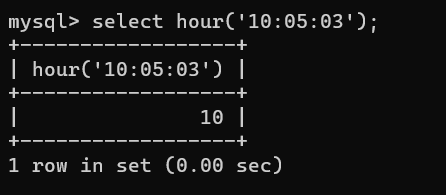


1. Hour()

* Extract the hour

Syntax :- [HOUR(***time***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_hour)

EX :-

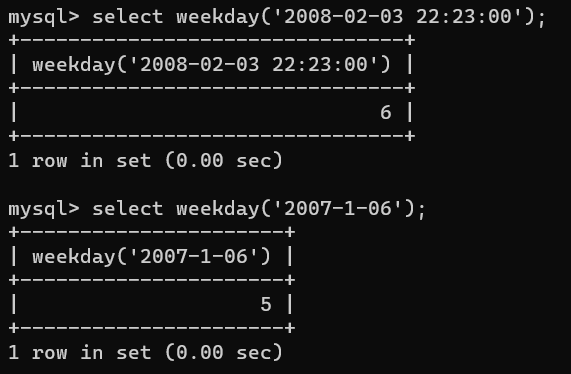


1. Weekday()

* Return the weekday index

Syntax :- [WEEKDAY(***date***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_weekday)

EX :-

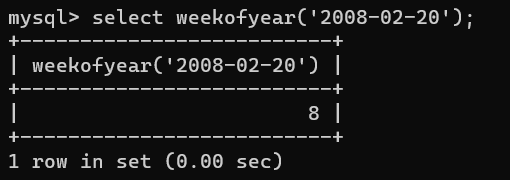


1. Weekofyear()

* Return the calendar week of the date (1-53)

Syntax :- [WEEKOFYEAR(***date***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_weekofyear)

EX :-

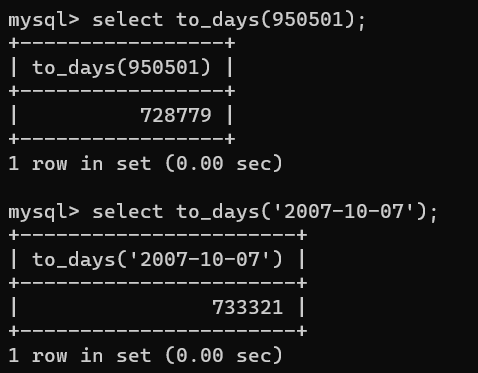


1. To\_days()

* Return the date argument converted to days

Syntax :- [TO\_DAYS(***date***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_to-days)

EX :-

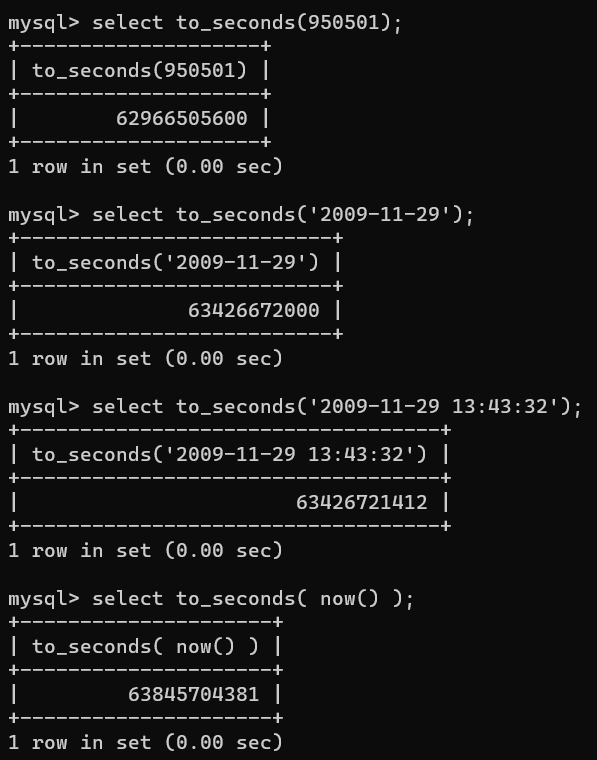


1. To\_seconds()

* Return the date or datetime argument converted to seconds since Year 0

Syntax :- [TO\_SECONDS(***expr***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_to-seconds)

EX :-

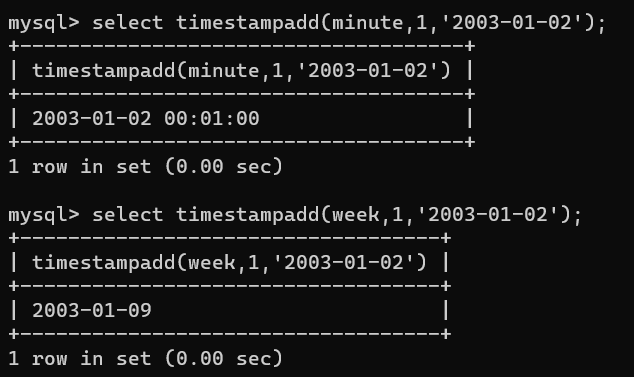


1. Timesstampadd()

* Add an interval to a datetime expression

Syntax :- [TIMESTAMPADD(***unit***,***interval***,***datetime\_expr***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_timestampadd)

EX :-

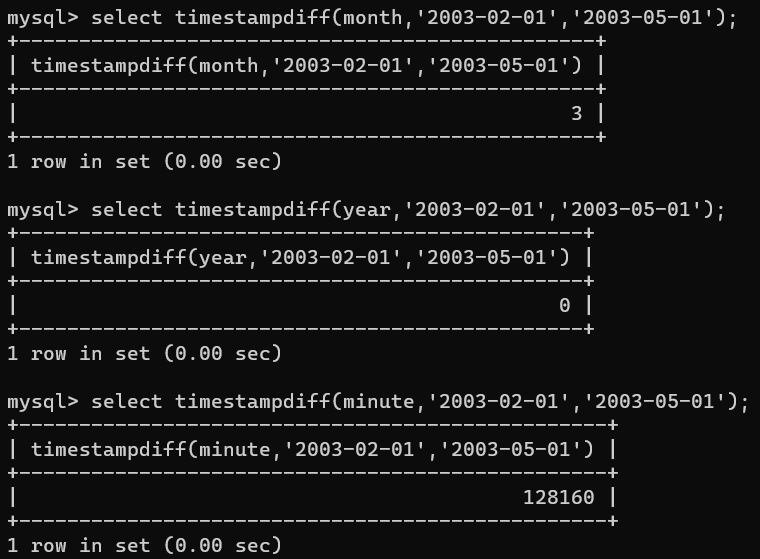


1. Timestampdiff()

* Return the difference of two datetime expressions, using the units specified

Syntax :- [TIMESTAMPDIFF(***unit***,***datetime\_expr1***,***datetime\_expr2***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_timestampdiff)

EX :-

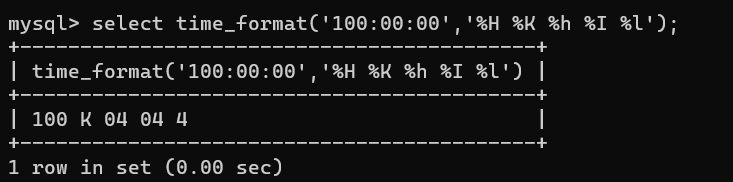


1. Time\_format()

* Format as time

Syntax :- [TIME\_FORMAT(***time***,***format***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_time-format)

EX :-

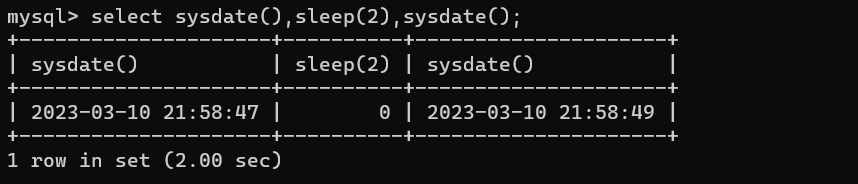


1. Sysdate()

* Return the time at which the function executes

Syntax :- [SYSDATE()](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_sysdate)

EX :-

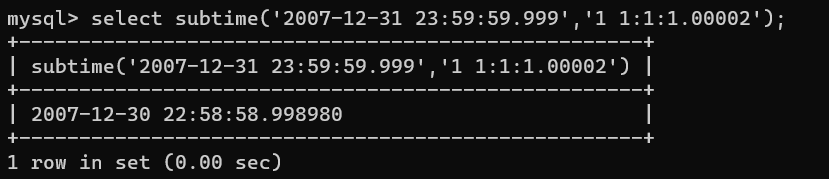


1. Subtime()

* Subtract times

Syntax :- [SUBTIME(***expr1***,***expr2***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_subtime)

EX :-



1. Subdate()

* Return the time at which the function executes

Syntax :- [SUBDATE(***date***,INTERVAL ***expr*** ***unit***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_subdate), [SUBDATE(***expr***,***days***)](https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html#function_subdate)

EX :-

